WIRELESS INTERNET SOFTWARE ENGINEERING IST-2000-30028

Title:
Service Engineering Process (Pilot Processes)

Version: 03.02
Date : 17 Sep 04
Pages : 100

Author(s):
Alexis Ocampo, Juergen Muench

To:
WISE CONSORTIUM

The WISE Consortium consists of:

Investnet, Motorola Technology Center Italy, Sodalia s.p.A, Sonera, Solid EMEA North, Fraunhofer IESE, Politecnico di Torino, VTT Electronics

Printed on:
17/09/2004 10.27

Status:

[ ] Draft
[ ] To be reviewed
[ ] Proposal
[X] Final/Released

Confidentiality:

[X] Public - Intended for public use
[ ] Restricted - Intended for WISE consortium only
[ ] Confidential - Intended for individual partner only

Deliverable ID: **D2 (Part C)**

Title:

Service Engineering Process (Pilot Processes)

Summary / Contents:

This document is part of deliverable D2, which describes the work done and results obtained for the WP1-Task: “Define a process to engineer services” of the WISE project. Deliverable D2 includes three parts: Part A: Service Engineering Process (Empirical approach for creating the reference process model), Part B: Service Engineering Process (The reference process model), and Part C: Service Engineering Process (Pilot Processes).
This part of the deliverable comprises process handbooks for the processes of Pilot 1, Pilot 2 client, and Pilot 2 server. These handbooks were used during the third iteration of the project in order to develop the respective pilot’s services. Additionally, electronic versions of the processes (so-called Electronic Process Guides) were provided and used by the pilot partners.

TABLE OF CONTENTS

CHANGE LOG

Vers.	Date	Author	Description
01.02	05/04/02	U. Becker-Kornstaedt, A Ocampo	Final version D2.V01.02
01.03	16/10/02	A. Ocampo, Dr. Jürgen Muench	Draft version D2.V01.03
01.03	12/11/02	A. Ocampo, Dr. Jürgen Muench	Final version D2.V01.03
01.04	08/01/03	A. Ocampo, Dr. Jürgen Muench	Final version D2.V01.04
02.01	06/04/03	A. Ocampo, Dr. Jürgen Muench	Draft version D2.V02.01
02.01	30/04/03	A. Ocampo, Dr. Jürgen Muench	Final version D2.V02.02
03.01	06/10/03	A. Ocampo, Dr. Jürgen Muench	Draft version D2.V03.01
03.02	23/10/03	A. Ocampo, Dr. Jürgen Muench	Final version D2.V03.02
05.01	23/06/04	A. Ocampo, Dr. Jürgen Muench	Draft version D2.V05.01

APPLICABLE DOCUMENT LIST

Ref.	Title, author, source, date, status	Identification
1	Indicators; Dr. Jürgen Münch; Fraunhofer IESE; 10/06/2003; Proposal	D8-V2
2	Analysis: Fabio Bella, Dr. Jürgen Münch, Alexis Ocampo; 09/06/2003	D9-V2
3	Heterogeneous Clients; Filippo Forchinno, Mario Negro Ponzi; 08/09/2003	D3-V2

1.1 MOTIVATION

The wireless Internet services domain is an upcoming new application domain, which can be characterized as follows: quickly evolving technology, upcoming new devices, new communication protocols, support for new different media types, varying and limited communication bandwidth, together with the need for new business models that will fit in with the completely new services portfolio. Examples of new Wireless Internet Services can be expected in the domain of Mobile Entertainment, Telemedicine, Travel Services, Tracking and Monitoring Services, or mobile trading services. At the moment, there is very little experience in developing software for such services systematically. Therefore, designing processes for this domain implicates several difficulties:

1. Whereas for conventional software development, several standards exist, for Wireless Internet Services no such standards are available that could be used as reference.
2. The Wireless Internet Services domain lacks specific experience on particular techniques, their applicability and constraints.
3. The variations of the applications and, as a consequence, possible variations of the development processes, are not sufficiently understood.
4. The impact of the variation of the enabling technology on the developed service is not always known and this may affect the development process.

There are several ways towards solving this problem: one widely accepted idea in the software engineering community is descriptive modeling of development processes, which leads to the explicit definition of process models, product models, and resource models.

Descriptive software process modeling captures processes as they take place in development. This approach was followed in order to describe the process models presented in this deliverable part (for more information on this approach please see part A section 3.1). Therefore, these Pilot process models are a reflection of the practices, processes followed by pilot partners in order to develop their correspondent wireless Internet services during the third iteration of the WISE project.

1.2 DOCUMENTATION

The deliverable (D2.V5) documents the methodology to create the reference process model. Part A presents the results of the step “Survey processes for software and system engineering”, “Define new process to develop wireless services”, “Define measures and indicators”, and “evaluate quality related activities”. Part B updates the reference process model as a result from the step “Define new process to develop wireless services, in terms of activities, techniques, tools, deliverables, and milestones“. Part C documents the pilots’ process models used for iteration 3 as a result of the step “Elicit existing process knowledge”. Please note in Table 1 that the documents’ version differs from the pilot’s process models version (PVX) and the Reference Process Model version (WISEPVX). This document corresponds to part C of the deliverable D2.V5.

Table 1. Documented results WP1 – Task 1.2

Process Models\Iteration	Start iteration 1	End iteration 1	Start iteration 2	End iteration 2	Start iteration 3	End iteration 3
Pilots 1 and pilot 2 planned processes	D2.V0 (PV0)		D2.V2 (PV1)		D2.V4 (PV2)	
Empirical approach for creating the Reference Process Model						D2.V5.A
Reference Process Model		D2.V1 (WISEPV1)	D2.V2 (WISEPV1)	D2.V3 (WISEPV2)	D2.V4 (WISEPV2)	D2.V5.B **(WISEPV3)**
Pilot 1 and pilot 2 actual processes		D2.V1 (PV1)		D2.V3 (PV2)		**D2.V5.C (PV3)**

2. PROCESS MODEL FOR PILOT 1: *ONLINE-TRADING*

This model describes the process followed by Pilot 1 during the third iteration of the WISE project.

2.1 OVERVIEW OF THE PROCESS MODEL

2.1.1 Activities

The following are the activities present in this process model:

- Requirements Phase
  - Elicit First Requirements
  - Code/Test Feasibility Study
  - Elicit Second Requirements
- Coding Phase
  - Coding Testing
- Testing Phase
  - Integration Testing
  - Documenting
  - Release System
  - Acceptance Testing

2.1.2 Artifacts

The following are the artifacts present in this process model:

- Request For Change
- Feasibility Study
- Requirements Specification
- Code
- Defect List
- Integrated Code
- Documentation
- System Released

2.1.3 Roles

The following are the roles present in this process model:

- Developer
- Manager
- Market & Trade Expert
- Pilot User
- Project Leader

2.1.4 Tools

The following are the tools present in this process model:

- Data Transfer Tool
- Mobile Device
- Sun One Studio
- Text Processor
- UML Editor
- WAP Emulator

2.1.5 Process View

The following is a global view for this process model:

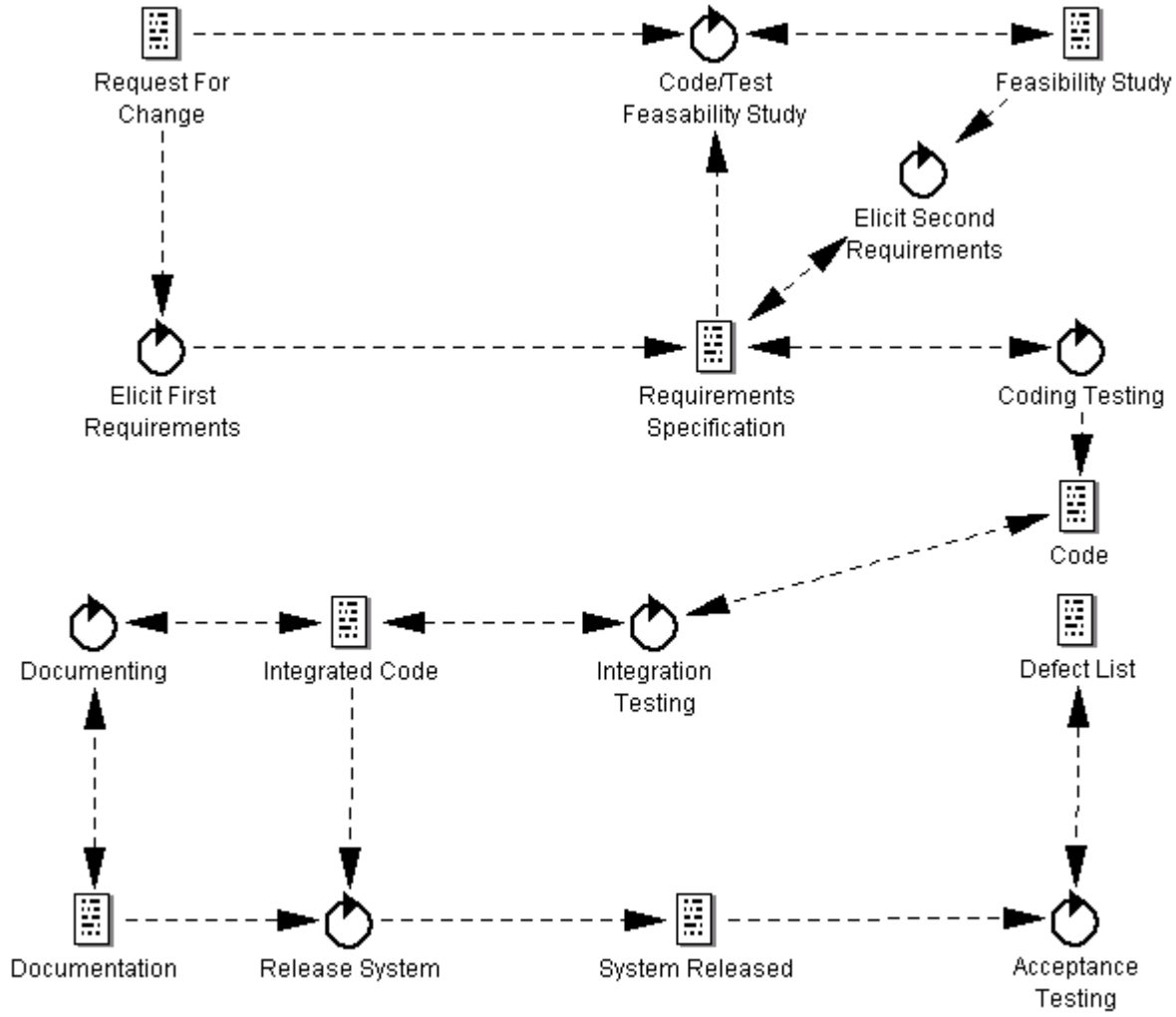

Figure 1. Global view.

2.1.6 Process View: Roles

The following is a global view of the roles for this process model:

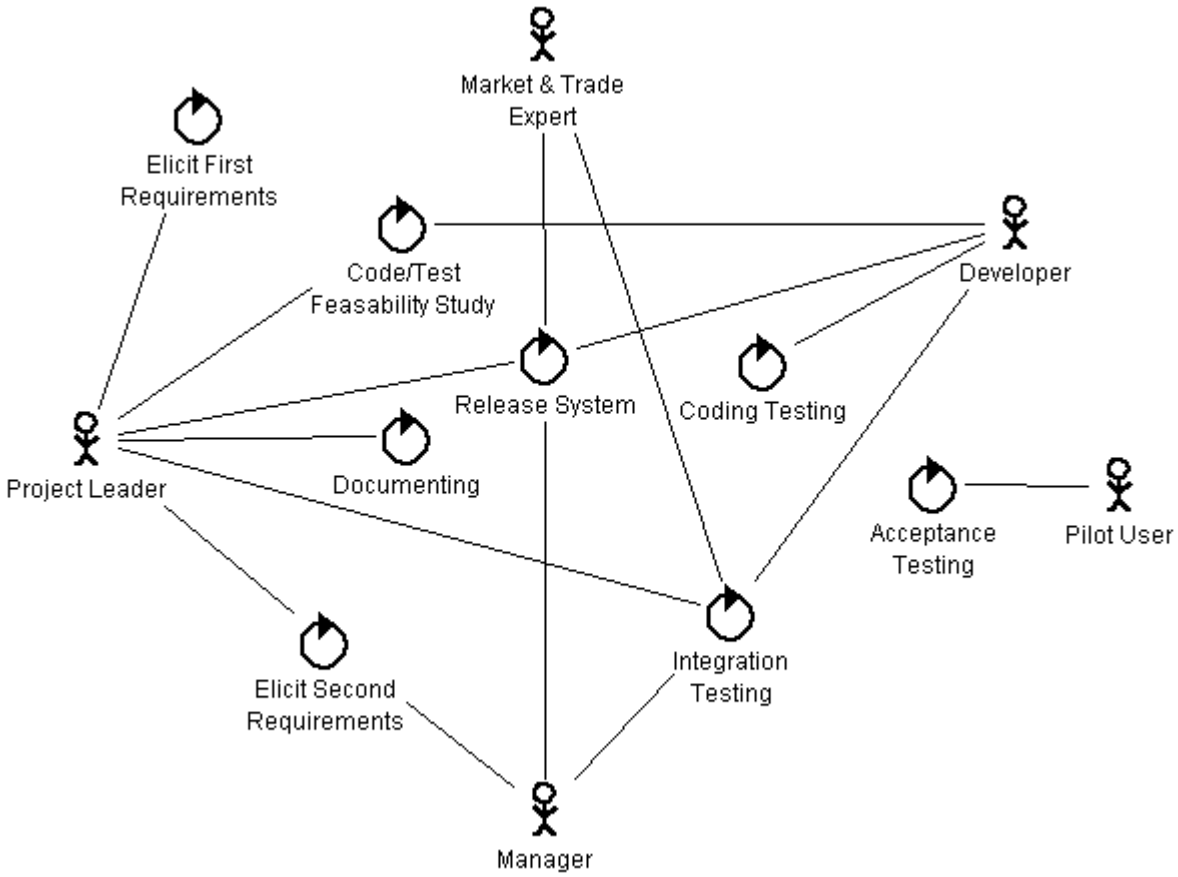

Figure 2. Process view: roles.

2.1.7 Process View: Tools

The following is a global view of the tools for this process model:

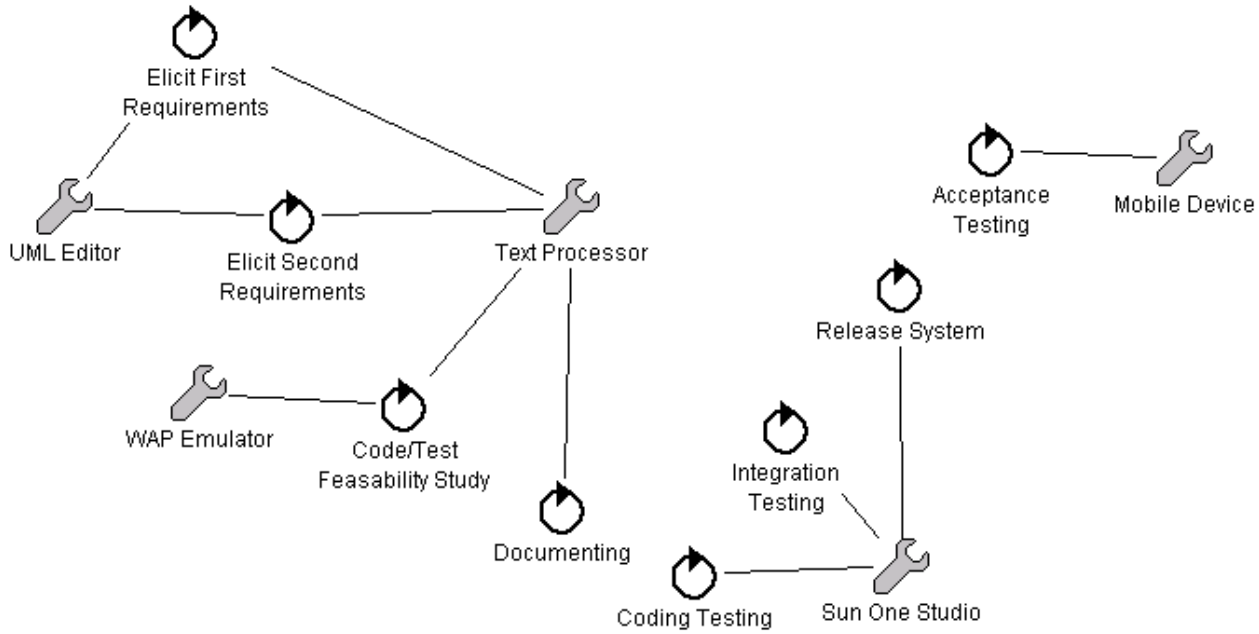

Figure 3. Process view: tools.

2.2 ONLINE-TRADING/ PHASE OVERVIEW

The following are the phases described in the process description:

- Requirements Phase
- Coding Phase
- Testing Phase

2.2.1 Product Flow Among Phases

The following graph depicts the product flow between the different phases:

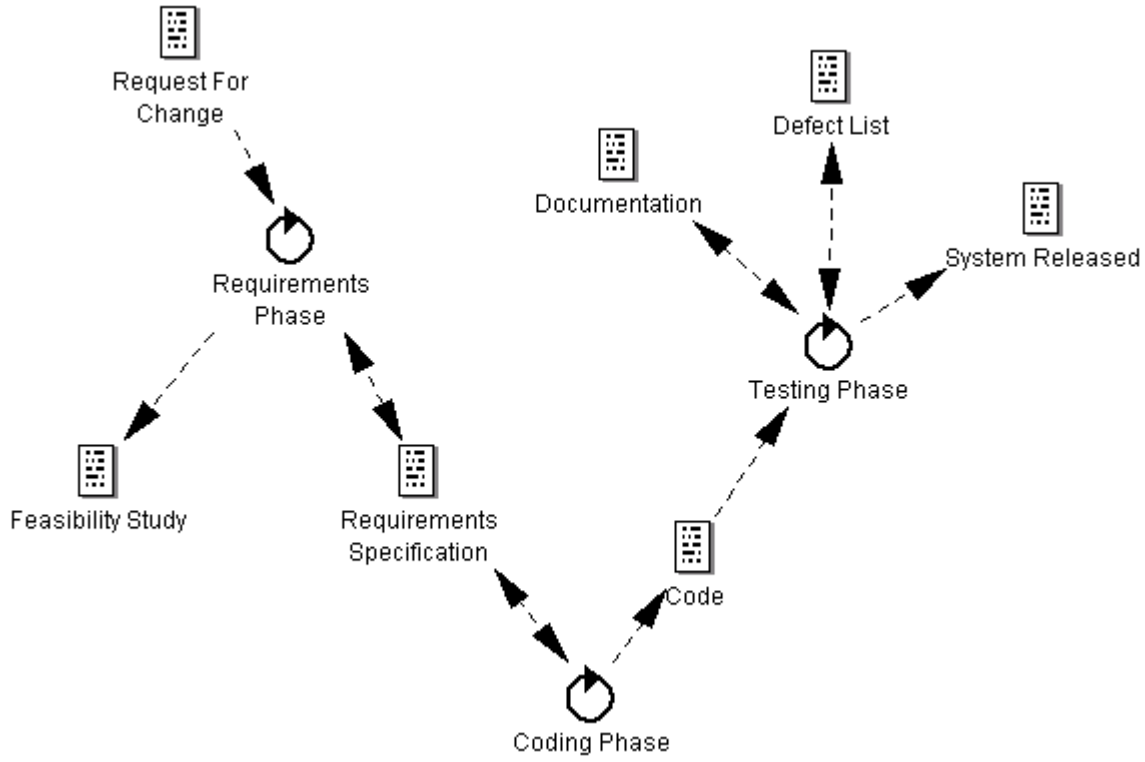

Figure 4. Product flow between phases.

2.3 ONLINE-TRADING/ REQUIREMENTS PHASE

2.3.1 Description

1. The requirements are gathered.
2. A feasibility study is performed.
3. The requirements are specified

2.3.2 Calendar Time

1. Days taken to complete the activities of this phase. (Begin date - End date)

2.3.3 Effort

1. Time spent in the set of activities for this phase. (Hours)

2.3.4 Input Criteria

1. The collected requirements from the customer were identified clearly as a new development from scratch or a modification of an existing product.
2. A defect from a previous version was identified and documented in the defect correction plan. The defect was traced back to a specified requirement. The requirement must be reworked.

2.3.5 Exit Criteria

1. A completed requirements specification.

2. The results of the feasibility study.
3. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

2.3.6 Product Flow

This activity consumes the following artifacts:
- Request For Change

This activity modifies the following artifacts:
- Requirements Specification

This activity produces the following artifacts:
- Feasibility Study

The following graph depicts the product flow for this phase:

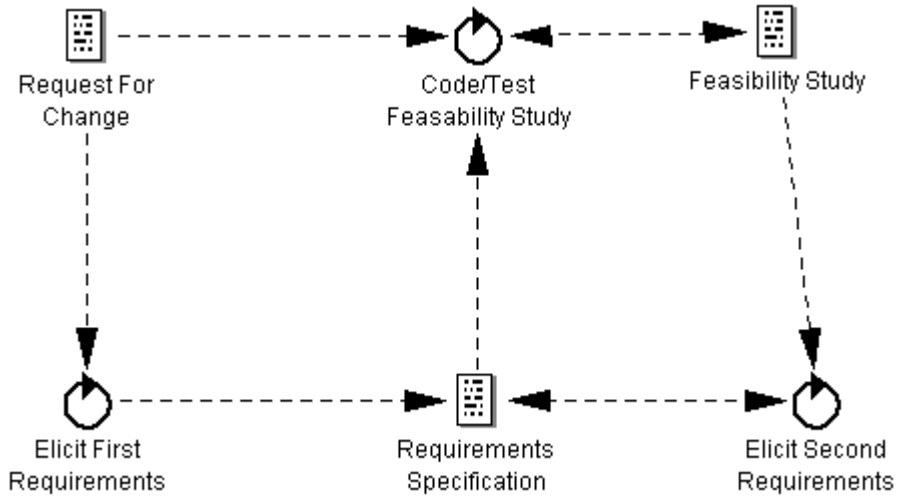

Figure 5. Product flow requirements phase.

2.3.7 Activities

The following are the activities for this phase:
- Elicit First Requirements
- Code/Test Feasibility Study
- Elicit Second Requirements

2.3.8 Activity: *Elicit First Requirements*

2.3.8.1 Description

Meeting to establish what could be implemented and what should be carried out as a feasibility study.
The feasibility study is often carried out on the following limitations
1. Select a set of mobile devices with the right display capabilities, and J2ME, (i.e., fundamental data of stock information. The idea is to avoid that user interaction on scrolling the data)
2. Look at network issues, and select a subset of devices.

2.3.8.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

2.3.8.3 Defects

Defects found during this process. Each defect has the following attributes:
- Defect type:
10. Documentation; 20. Syntax; 30. Build, Package;40. Assignment;50. Interface;60. Checking;70. Data;80. Function;90. System;100. Environment;110. Others.
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.
- Fix time
- Comments

2.3.8.4 Effort

1. Time spent producing the first draft of the requirements document. (Hours)

2.3.8.5 Input Criteria

1. The collected requirements from the customer were identified clearly as a new development from scratch or a modification of an existing product.
2. A defect from a previous version was identified and documented in the defect correction plan. The defect was traced back to a specific requirement. The requirement must be reworked.

2.3.8.6 Exit Criteria

1. The first draft of the requirements document is provided.
2. The information was approved by the customer.
3. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

2.3.8.7 Product Flow

This activity consumes the following artifacts:
- Request For Change

This activity does not modify any artifacts.
This activity produces the following artifacts:
- Requirements Specification

2.3.8.8 Involved Roles

The following roles are involved with this activity:
- Project Leader

2.3.8.9 Used Tools

This activity uses the following tools:
- Text Processor
- UML Editor

2.3.9 Activity: *Code/Test Feasibility Study*

2.3.9.1 Description

Doubts concerning new technologies are resolved.
Prototypes are developed for verifying what is feasible.

2.3.9.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

2.3.9.3 Defects

Defects found during this process. Each defect has the following attributes:
- Defect type: 10. Documentation; 20. Syntax; 30. Build, Package;40. Assignment;50. Interface;60. Checking;70. Data;80. Function;90. System;100. Environment;110. Others.
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.
- Fix time
- Comments

2.3.9.4 Effort

1. Time spent studying feasibility. (Hours)

2.3.9.5 Input Criteria

1. The collected requirements from the customer were identified clearly as a new development from scratch or a modification of an existing product.
2. A defect from a previous version was identified and documented in the defect correction plan. The defect was traced back to an specific requirement. The requirement must be reworked.
3. A previous feasibility study that needs to be updated.
4. Approved first draft of the requirements document.

2.3.9.6 Exit Criteria

1. A completed feasibility study
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

2.3.9.7 Product Flow

This activity consumes the following artifacts:
- Request For Change
- Requirements Specification

This activity modifies the following artifacts:
- Feasibility Study

This activity does not produce any artifacts.

2.3.9.8 Involved Roles

The following roles are involved with this activity:
- Developer
- Project Leader

2.3.9.9 Used Tools

This activity uses the following tools:
- Text Processor
- WAP Emulator

2.3.10 Activity: *Elicit Second Requirements*

2.3.10.1 Description

Meeting to decide which functionality can be implemented. In this meeting the development plan is defined. The results are approved by the manager and requirements engineer

2.3.10.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

2.3.10.3 Defects

Defects found during this process. Each defect has the following attributes:
- Defect type:
10. Documentation; 20. Syntax; 30. Build, Package;40. Assignment;50. Interface;60. Checking;70. Data;80. Function;90. System;100. Environment;110. Others.
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.
- Fix time
- Comments

2.3.10.4 Effort

1. Time spent completing requirements specification document. (Hours)

2.3.10.5 Input Criteria

1. Approved first draft of the requirements document.

2.3.10.6 Exit Criteria

1. A completed requirements specification document.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

2.3.10.7 Product Flow

This activity consumes the following artifacts:
- Feasibility Study

This activity modifies the following artifacts:
- Requirements Specification

This activity does not produce any artifacts.

2.3.10.8 Involved Roles

The following roles are involved with this activity:
- Manager
- Project leader

2.3.10.9 Used Tools

This activity uses the following tools:
- Text Processor
- UML Editor

2.4 *ONLINE-TRADING/ CODING PHASE*

2.4.1 Description

1. The code is produced and tested.
2. The code is integrated.
3. The code is released.

2.4.2 Calendar Time

1. Days taken to complete the activities of this phase. (Begin date - End date)

2.4.3 Effort

1. Time spent in the set of activities for this phase. (Hours)

2.4.4 Input Criteria

1. A completed requirements specification.

2.4.5 Exit Criteria

1. Code that complies with the coding standards.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

2.4.6 Product Flow

This activity does not consume any artifacts.
This activity modifies the following artifacts:

- Requirements Specification

This activity produces the following artifacts:

- Code

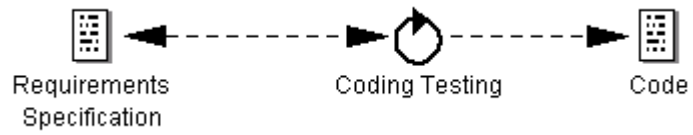

Figure 6. Product flow development phase.

2.4.7 Activities

The following are the activities of this phase:

- Coding Testing

2.4.8 Activity: *Coding Testing*

2.4.8.1 Description

1. Produce the units of code (modules, functions, methods) that complies with the pilot functionality.

2.4.8.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

2.4.8.3 Defects

Defects found during this process. Each defect has the following attributes:

- Defect type:
10. Documentation; 20. Syntax; 30. Build, Package;40. Assignment;50. Interface;60. Checking;70. Data;80. Function;90. System;100. Environment;110. Others.
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.
- Fix time
- Comments

2.4.8.4 Effort

1. Time spent coding (modules, functions, methods). (Hours).

2.4.8.5 Input Criteria

1. A completed requirements specification.

2.4.8.6 Exit Criteria

1. Code that complies with the coding standards.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

2.4.8.7 Product Flow

This activity does not consume any artifacts.
This activity modifies the following artifacts:

- Requirements Specification

This activity produces the following artifacts:

- Code

2.4.8.8 Involved Roles

The following roles are involved with this activity:

- Developer

2.4.8.9 Used Tools

This activity uses the following tools:

- Data Transfer Tool
- Sun One Studio

2.5 *ONLINE-TRADING/ TESTING PHASE*

2.5.1 Description

This phase summarizes all activities related to testing.

2.5.2 Calendar Time

1. Days taken to complete the activities of this phase. (Begin date - End date)

2.5.3 Effort

1. Time spent in the set of activities for this phase. (Hours)

2.5.4 Input Criteria

1. Code that complies with the coding standards.

2.5.5 Exit Criteria

1. Completed user and technical manuals.
2. Reported problems. (e.g., via Web)
3. Completed system and customized variations.
4. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

2.5.6 Product Flow

This activity consumes the following artifacts:

- Code

This activity modifies the following artifacts:

- Defect List
- Documentation

This activity produces the following artifacts:

- System Released

2.5.7 Activities

The following are the activities of this phase:

- Integration Testing
- Documenting
- Release System
- Acceptance Testing

2.5.8 Product Flow Refinement

The following graph(s) depict the product flow refinement for this phase:

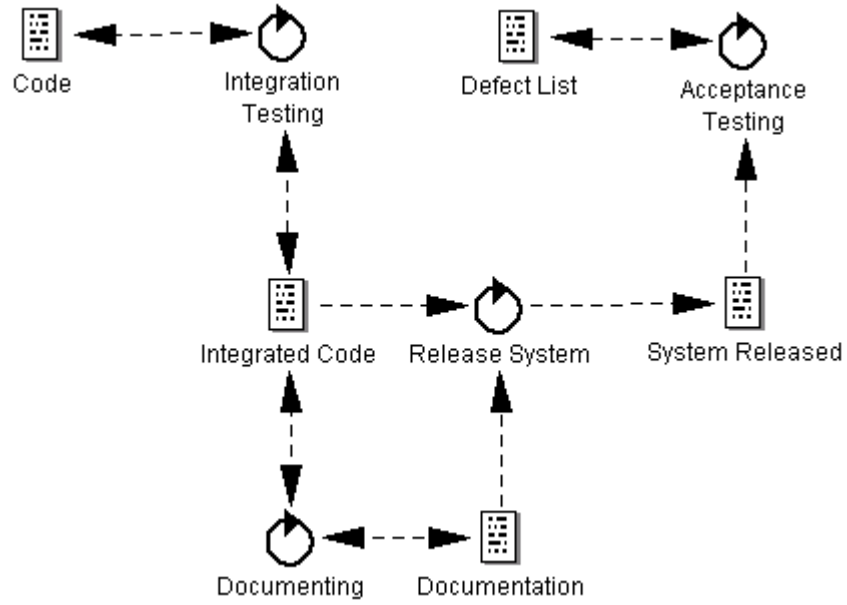

Figure 7. Product flow test phase.

2.5.9 Activity: *Integration Testing*

2.5.9.1 Description

1. Verify that all needed parts are on hand.
2. Build the product.
3. The system is tested by developers and market experts. Defects found are reported informally. Suggestions are also reported.
4. Every defect found is fixed.
5. The demo is packaged into the device and submitted to the market division for further tests.

2.5.9.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

2.5.9.3 Defects

Defects found during this process. Each defect has the following attributes:
- Defect type:
10. Documentation; 20. Syntax; 30. Build, Package;40. Assignment;50. Interface;60. Checking;70. Data;80. Function;90. System;100. Environment;110. Others
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.
- Fix time
- Comments

2.5.9.4 Effort

Time spent integrating the code (Hours).
Time spent deploying the code in the hardware and network platforms. (Hours)

2.5.9.5 Input Criteria

1. Code that complies with the coding standards.

2.5.9.6 Exit Criteria

1. Integrated code that complies with the coding standards.
2. Deployed code in the hardware and network platforms.
3. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

2.5.9.7 Product Flow

This activity does not consume any artifacts.
This activity modifies the following artifacts:
- Code
- Integrated Code

This activity does not produce any artifacts.

2.5.9.8 Involved Roles

The following roles are involved with this activity:
- Developer
- Manager

- Market & Trade Expert
- Project Leader

2.5.9.9 Used Tools

This activity uses the following tools:
- Data Transfer Tool
- Sun One Studio

2.5.10 Activity: *Documenting*

2.5.10.1 Description

1. Document the application functionality. (User Manual)
2. Document technical aspects (installation, maintenance, recovery, security)

2.5.10.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

2.5.10.3 Defects

Defects found during this process. Each defect has the following attributes:
- Defect type:10. Documentation; 20. Syntax; 30. Build, Package; 40. Assignment; 50. Interface; 60. Checking; 70. Data; 80. Function; 90. System; 100. Environment; 110. Others.
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.
- Fix time
- Comments

2.5.10.4 Effort

1. Time spent completing the user and technical manuals. (Hours)

2.5.10.5 Input Criteria

1. Integrated code that complies with the coding standards.
2. Deployed code in the hardware and network platforms.

2.5.10.6 Exit Criteria

1. Completed user and technical manuals.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

2.5.10.7 Product Flow

This activity does not consume any artifacts.
This activity modifies the following artifacts:
- Integrated Code
- Documentation

This activity does not produce any artifacts.

2.5.10.8 Involved Roles

The following roles are involved with this activity:

- Project Leader

2.5.10.9 Used Tools

This activity uses the following tools:

- Text Processor

2.5.11 Activity: *Release System*

2.5.11.1 Description

1. Review that the system is accepted by the market division.
2. Code customization details according to the final customer.
3. Package final version of the system.

2.5.11.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

2.5.11.3 Defects

Defects found during this process. Each defect has the following attributes:
- Defect type: 10. Documentation; 20. Syntax; 30. Build, Package;40. Assignment;50. Interface;60. Checking;70. Data;80. Function;90. System;100. Environment;110. Others
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.
- Fix time
- Comments

2.5.11.4 Effort

1. Time spent releasing the system and customized variations. (Hours).

2.5.11.5 Input Criteria

1. Integrated code that complies with the coding standards.
2. Deployed code in the hardware and network platforms.
3. Completed user and technical manuals.

2.5.11.6 Exit Criteria

1. Completed system and customized variations.

2.5.11.7 Product Flow

This activity consumes the following artifacts:

- Integrated Code
- Documentation

This activity does not modify any artifacts.
This activity produces the following artifacts:

- System Released

2.5.11.8 Involved Roles

The following roles are involved with this activity:

- Developer
- Manager
- Market & Trade Expert
- Project Leader

2.5.11.9 Used Tools

This activity uses the following tools:

- Data Transfer Tool
- Sun One Studio

2.5.12 Activity: *Acceptance Testing*

2.5.12.1 Description

1. The final customer tests the system and reports problems

2.5.12.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

2.5.12.3 Defects

Defects found during this process. Each defect has the following attributes:
- Defect type: 10. Documentation; 20. Syntax; 30. Build, Package;40. Assignment;50. Interface;60. Checking;70. Data;80. Function;90. System;100. Environment;110. Others
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.
- Fix time
- Comments

2.5.12.4 Effort

1. Time spent performing the acceptance tests of the system. (Hours)

2.5.12.5 Input Criteria

1. Completed system and customized variations.

2.5.12.6 Exit Criteria

1. Reported problems. (e.g., via Web)
2. System tested and accepted.

2.5.12.7 Product Flow

This activity consumes the following artifacts:

- System Released

This activity modifies the following artifacts:

- Defect List

This activity does not produce any artifacts.

2.5.12.8 Involved Roles

The following roles are involved with this activity:

- Pilot User

2.5.12.9 Used Tools

This activity uses the following tools:

- Mobile Device

2.6 *ONLINE-TRADING/ ARTIFACTS*

2.6.1 Artifact: *Request For Change*

2.6.1.1 Description

New ideas for the next version are collected from the developers and customer experiences, the market trends, and the technological changes. Thus a request to update/enhance/change the existing system is created. Bugs and features are also taken from the results of the previous system tested which can be seen in the defect list.

2.6.1.2 Product Flow

This artifact is not produced by any activity.
This artifact is not modified by any activity.
This artifact is used by the following activities:

- Requirements Phase
- Elicit First Requirements
- Code Test Feasibility Study

2.6.1.3 Size

Other measures suggested:
1. Number of pages

2.6.2 Artifact: *Feasibility Study*

2.6.2.1 Description

Structure of the document
SCOPE
The scope of this document is to describe the major responsibilities of the Server side and of the Client side for each service offered by the Pilot, in order to understand if the implementation is feasible in the defined timeframe.
BASIC FEATURES OF THE SERVICE
Describe the main characteristics of the service
CLIENT RESPONSABILITIES
List of the high-level client requirements
SERVER RESPONSABILITIES
List of the high-level server requirements
OBSERVATIONS
Performance requirements or constraints of the service
PROBLEMS
- Problem Id: Number
- Context: WISE Project Number of the pilot

- Description: Problem description
- Cause: Description of the problem's reason.
- Solution (Reactive)
- Solution (Preventive)
- References: Links to other similar problems and solutions
- Additional documentation: Links to information that clarifies the technical background of the problem.

CONCLUSIONS
The feasibility study concludes whether the implementation of the service is feasible in the defined timeframe. Guidelines could be extracted from the solutions of the problems
Unstructured document:
Informal description of the feasibility study

2.6.2.2 Product Flow

This artifact is produced by the following activities:
- Requirements Phase

This artifact is modified by the following activities.
- Code/Test Feasibility Study

This artifact is used by the following activities:
- Elicit Second Requirements

2.6.3 Artifact: *Requirements Specification*

2.6.3.1 Description

Detailed description of the requirements that were investigated and that will be implemented, and a basic plan of development.

2.6.3.2 Product Flow

This artifact is produced by the following activities:
- Elicit First Requirements

This artifact is modified by the following activities.
- Requirements Phase
- Elicit Second Requirements
- Coding Phase
- Coding Testing

This artifact is used by the following activities:
- Code/Test Feasibility Study

2.6.3.3 Size

Other measures suggested:
1. Number of pages

2.6.4 Artifact: *Code*

2.6.4.1 Description

Completed HTML code plus the server side code and the scripting client code.

2.6.4.2 Product Flow

This artifact is produced by the following activities:
- Coding Phase
- Coding Testing

This artifact is modified by the following activities.
- Integration Testing

This artifact is used by the following activities:
- Testing Phase

2.6.4.3 Size

1. Number of implemented functions

Other measures suggested:

2. Lines of code
3. Number of asp pages implemented

2.6.5 Artifact: *Defect List*

2.6.5.1 Description

Lists all defects found during the acceptance test performed by the friendly customer and the developers.

2.6.5.2 Product Flow

This artifact is not produced by any activity.
This artifact is modified by the following activities.
- Testing Phase
- Acceptance Testing

This artifact is not used by any activity.

2.6.5.3 Size

Other measures suggested:

1. Number of defects found at the acceptance test.

2.6.6 Artifact: *Integrated Code*

2.6.6.1 Description

Preliminary version of the system. The system has been tested and deployed. The system has been considered to be free of defects and ready to be presented to the market division.

2.6.6.2 Product Flow

This artifact is not produced by any activity.
This artifact is modified by the following activities.
- Integration Testing
- Documenting

This artifact is used by the following activities:
- Release System

2.6.7 Artifact: *Documentation*

2.6.7.1 Description

Technical and user manual. For demos or small applications brief documentation that describes the functionality and technical issues is provided. On the other hand, for commercial applications detailed user and technical manuals are written. Information is provided as CD, FTP.

2.6.7.2 Product Flow

This artifact is not produced by any activity.
This artifact is modified by the following activities.

- Testing Phase
- Documenting

This artifact is used by the following activities:

- Release System

2.6.8 Artifact: *System Released*

2.6.8.1 Description

Final version of the system. The system has been tested, deployed, and accepted by the marketing division. The system has been considered to be free of defects and customized for a final customer.

2.6.8.2 Product Flow

This artifact is produced by the following activities:

- Testing Phase
- Release System

This artifact is not modified by any activity.
This artifact is used by the following activities:

- Acceptance Testing

2.7 *ONLINE-TRADING/ ROLES*

2.7.1 Role: D*eveloper*

2.7.1.1 Description

Writes down the code for the feasibility study prototype.

2.7.1.2 Participation

This role participates in the following activities:

- Code/Test Feasibility Study
- Coding Testing
- Integration Testing
- Release System

2.7.2 Role: *Manager*

2.7.2.1 Description

Reviews the first draft of the requirements specification.

2.7.2.2 Participation

This role participates in the following activities:

- Elicit Second Requirements
- Integration Testing
- Release System

2.7.3 Role: *Market & Trade Expert*

2.7.3.1 Description

Traders provide feedback on the application. He can perform benchmarking with other applications. He gives opinions on usability of the application from a trader point of view. He gives opinion on the logical sequence of functionality.
Commercial experts verify if the application matches the desires of the customer.
Performance is checked by this profile

2.7.3.2 Participation

This role participates in the following activities:

- Integration Testing
- Release System

2.7.4 Role: *Pilot User*

2.7.4.1 Description

Final friendly user in charge of providing feedback about the problems found when doing acceptance test or suggestions to improve the product.

2.7.4.2 Participation

This role participates in the following activities:

- Acceptance Testing

2.7.5 *Role*: *Project Leader*

2.7.5.1 Description

In charge of writing and reviewing the requirements specification
He leads the development of the prototype for the feasibility study. He takes technical implementation decisions or answer questions regarding the technical decisions of the prototype.

2.7.5.2 Participation

This role participates in the following activities:

- Elicit First Requirements
- Code/Test Feasibility Study
- Elicit Second Requirements
- Integration Testing
- Documenting
- Release System

2.8 *ONLINE-TRADING/ TOOLS*

2.8.1 Tool: *Data Transfer Tool*

2.8.1.1 Usage

This tool is used by the following activities:

- Coding Testing
- Integration Testing
- Release System

2.8.2 Tool: *Mobile device*

2.8.2.1 Usage

This tool is used by the following activities:

- Acceptance Testing

2.8.3 Tool: *Sun One Studio*

2.8.3.1 Usage

This tool is used by the following activities:

- Coding Testing
- Integration Testing
- Release System

2.8.4 Tool: *Text Processor*

2.8.4.1 Usage

This tool is used by the following activities:

- Elicit First Requirements
- Code/Test Feasibility Study
- Elicit Second Requirements
- Documenting

2.8.5 Tool: *UML Editor*

2.8.5.1 Usage

This tool is used by the following activities:

- Elicit First Requirements
- Elicit Second Requirements

2.8.6 Tool: *WAP Emulator*

2.8.6.1 Description

To verify the display capabilities

2.8.6.2 Usage

This tool is used by the following activities:

- Code/Test Feasibility Study

3. PROCESS MODEL FOR PILOT 2 CLIENT: *ONLINE-ENTERTAINMENT*

This model describes the process followed by Pilot 2 Client during the third iteration of the WISE project. The information in the model is based mainly on examples of the processes as they have been performed in the respective environments.

3.1 OVERVIEW OF THE PROCESS MODEL

3.1.1 Activities

The following are the activities present in this process model:

- Requirements Phase
  - Gather Requirements
    - Analyze User Interface Feasibility Requirements
    - Identify Scenarios
    - Specify Initial Requirements
  - feasibility_study
    - Search Possible Solutions
    - Test Possible Solutions
  - Analyze Requirements
- Design Phase
  - System Design
  - Design
  - Design Review
- Coding Phase.
  - Code
  - Unit Test
  - Integrate Code
  - Release Code
- Testing Phase.
  - Plan Tests
  - Build Test Framework
  - Test System
  - Test Usability
  - Acceptance Test
  - Analyze Defect

3.1.2 Artifacts

The following are the artifacts in this process model:

- Requests From Customer
- SRS
- List Possible Solutions
- Possible Scenarios
- Feasibility Study
- Architecture Template
- Defect Log
- Design Document
- Design Review Checklist
- Source Code
- Integrated Code
- Code Released
- Test Plan

- Test Framework
- RQTM
- Product Tested
- Usability Interview
- Usability Report
- Test Report

3.1.3 Roles

The following are the roles present in this process model:

- Customer
- Developer
- Project Leader
- Project Manager
- Technical Leader

3.1.4 Tools

The following are the tools present in this process model:

- J2EE
- J2ME
- UML Editor
- Configuration Management Tool
- Integrated Development Environment
- Real Mobile Device
- Text Editor

3.1.5 Process View

The following is a global view for this process model:

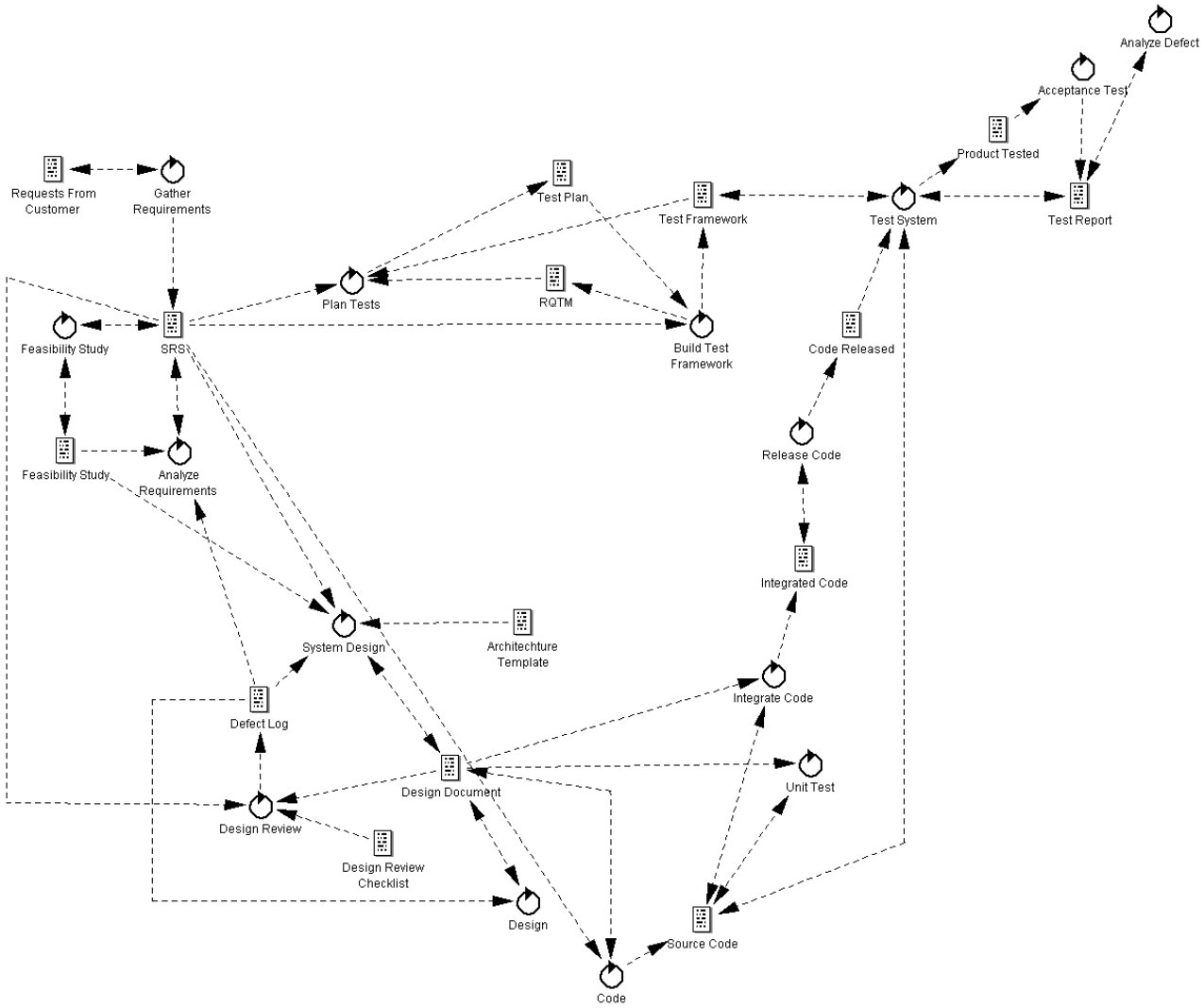

Figure 8. Global view.

3.2 ONLINE-ENTERTAINMENT CLIENT /PHASE_OVERVIEW

The following are the phases described in this process description:

- Requirements Phase.
- Design Phase.
- Coding Phase.
- Testing Phase.

3.2.1 Product Flow Among Phases

The following graph depicts the product flow between the different phases.

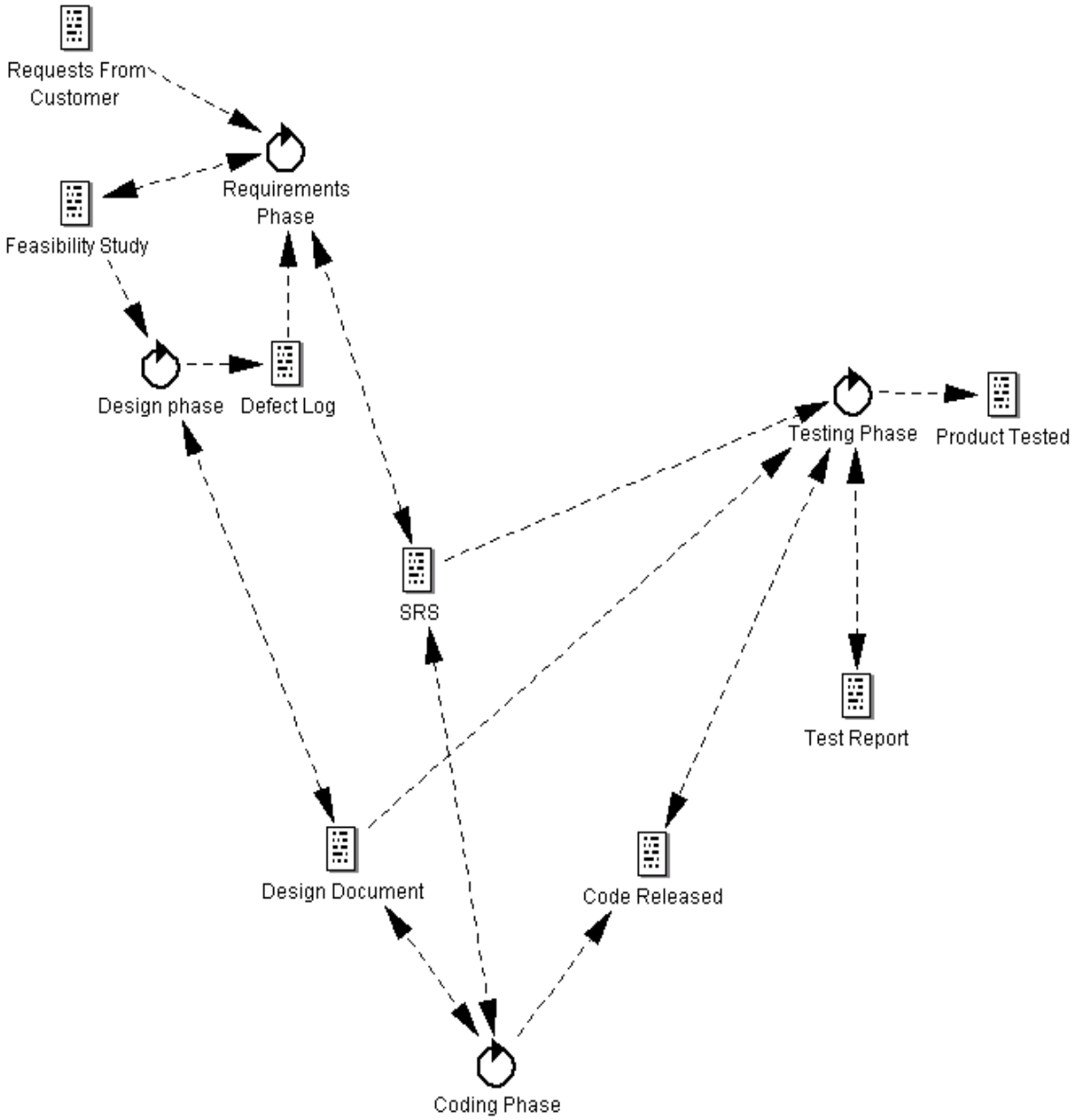

Figure 9. Product flow between phases.

3.3 ONLINE-ENTERTAINMENT CLIENT /REQUIREMENTS PHASE

3.3.1 Description

1. The requirements are gathered.
2. A feasibility study is performed.
3. The requirements are specified.

3.3.2 Calendar Time

1. Days taken to complete the activities of this phase. (Begin date - End date)

3.3.3 Effort

1. Time spent in the set of activities for this phase. (Hours)

3.3.4 Input Criteria

1. The collected requirements from the customer were identified clearly as a new development from scratch or a modification of an existing product.
2. A defect from a previous version was identified and documented in the defect correction plan. The defect was traced back to a specified requirement. The requirement must be reworked.

3.3.5 Exit Criteria

1. A completed requirements specification.
2. The results of the feasibility study.
3. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

3.3.6 Product Flow

This activity consumes the following artifacts:

- Requests From Customer
- Defect Log

This activity modifies the following artifacts:

- SRS
- Feasibility Study

This activity does not produce any artifacts.

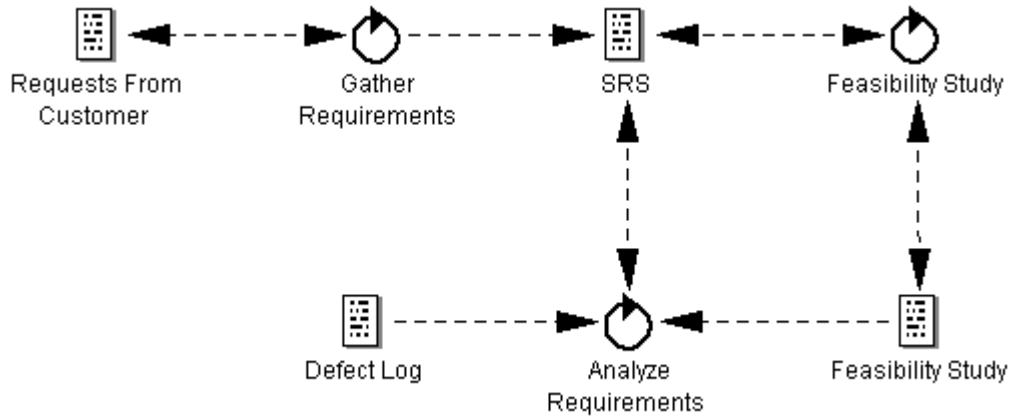

Figure 10. Product flow requirements phase.

3.3.7 Process View: Roles

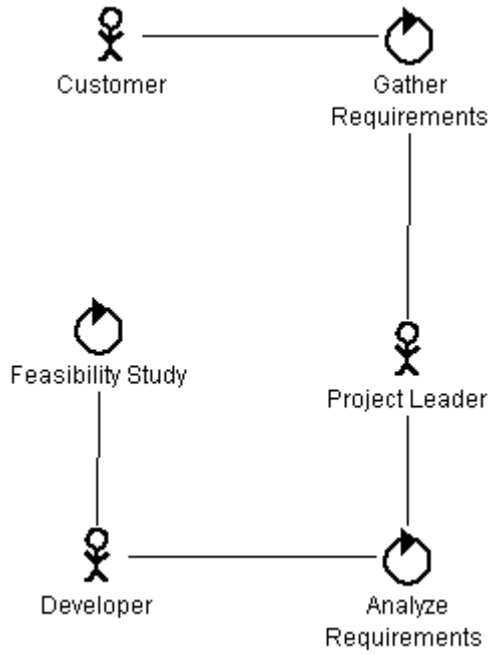

Figure 11. Process view: roles.

3.3.8 Process View: Tools

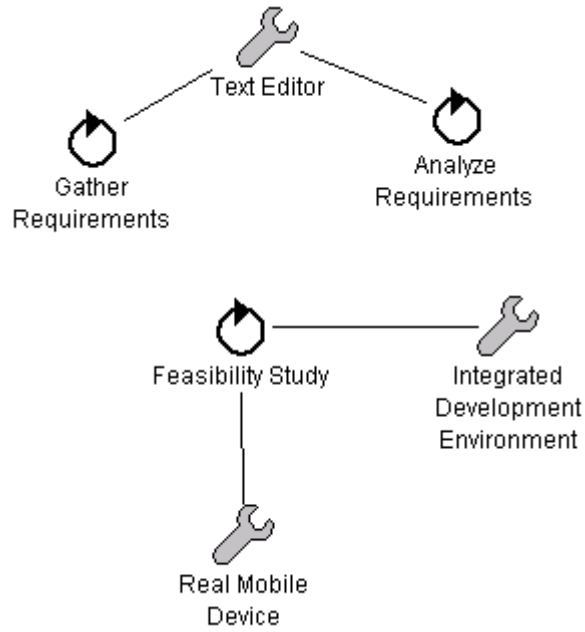

Figure 12. Process view: tools.

3.3.9 Activities

The following are the subactivities of this phase:
- Gather Requirements
- Feasibility Study
- Analyze Requirements

3.3.10 Activity: *Gather Requirements*

3.3.10.1 Description

1. Collect requirements from customer. Customer may give an informal document containing the requirements.
2. Ensure the requirements from the customer are clear and unambiguous.
3. Resolve any questions.
4. Produce requirements document: It shall be approved by the customer.

3.3.10.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.3.10.3 Effort

1. Time spent producing the first draft of the requirements document (Hours).

3.3.10.4 Input Criteria

1. The collected requirements from the customer were identified clearly as a new development from scratch or a modification of an existing product.

3.3.10.5 Exit Criteria

1. The first draft of the requirements document is provided.
2. The information was approved by the customer.

3.3.10.6 Product Flow

This activity does not consume any artifacts.
This activity modifies the following artifacts:
- Requests From Customer

This activity produces the following artifacts:
- SRS

3.3.10.7 Activities

The following are the subactivities of this activity:
- Analyze User Interface Feasibility Requirements
- Identify Scenarios
- Specify Initial Requirements

3.3.10.8 Product Flow Refinement

The following graph(s) depict the product flow refinement for this activity.

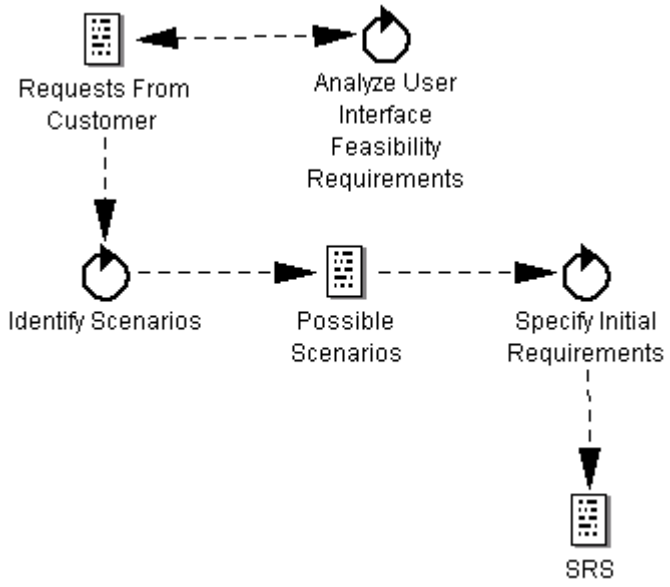

Figure 13. Product flow gather requirements.

3.3.10.9 Process View: Roles

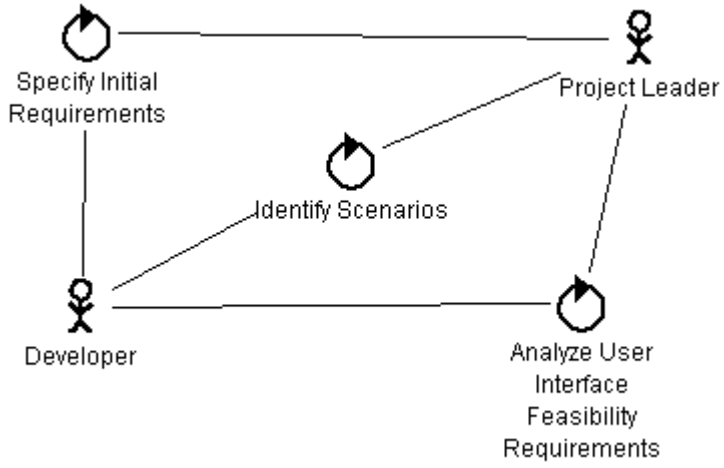

Figure 14. Process view: gather requirements roles.

3.3.10.10 Process View: Tools

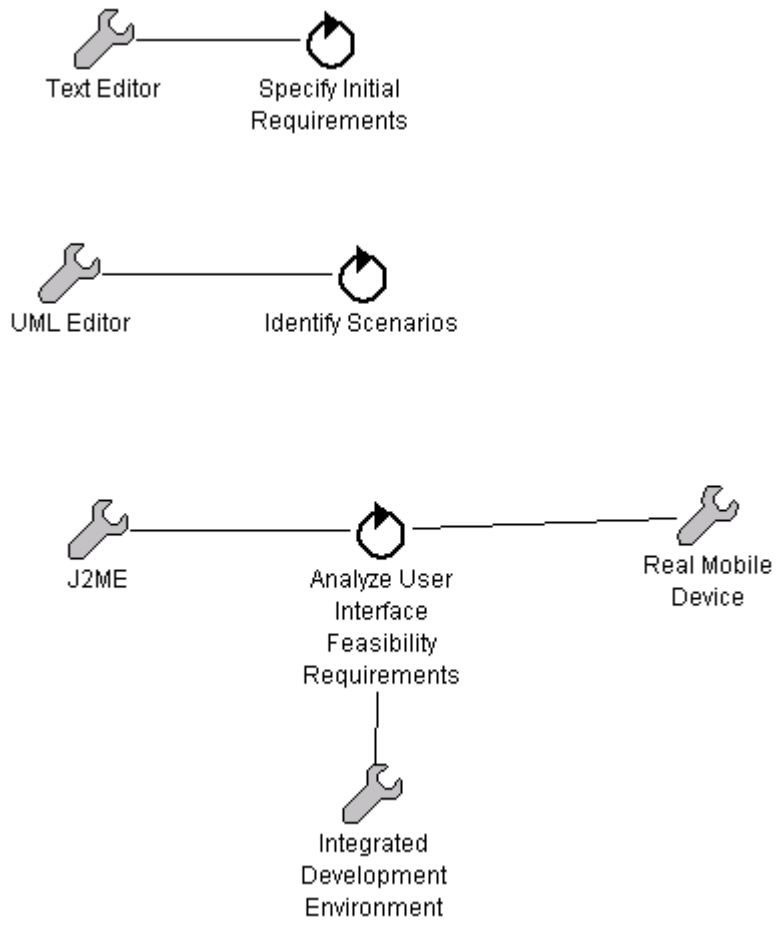

Figure 15. Process view: gather requirements tools.

3.3.10.11 Involved Roles

The following roles are involved with this activity:

- Customer
- Project Leader
- Technical Leader

3.3.10.12 Used Tools

This activity uses the following tools:

- Text Editor

3.3.10.13 Activity: Analyze User Interface Feasibility Requirements

3.3.10.13.1 Description

The manipulation and implementation of the graphical user interface in mobile devices is at an early stage. Open source graphic libraries are still far from being accepted as a standard for commercial releases. Therefore, technology providers have created proprietary libraries that extend the capabilities of existing software and allows them to manipulate better its own technology (devices). Pilot 2, has such libraries and

they drive the capabilities of their user interfaces. Every time a new description of the service arrives, developers must validate the wishes of the customer with the existent capabilities. Once this analysis is done, a decision on accepting the user interface requirements or rejecting them is made. Sometimes, developers are not sure about the feasibility of requirements. These requirements are partially received but research deeply in a feasibility study.

3.3.10.13.2 Input Criteria

1. The collected requirements from the customer were identified clearly as a new development from scratch or a modification of an existing product.

3.3.10.13.3 Exit Criteria

1. An updated request from customer.

3.3.10.13.4 Product Flow

This activity does not consume any artifacts.
This activity modifies the following artifacts:

- Requests From Customer

This activity does not produce any artifacts.

3.3.10.13.5 Involved Roles

The following roles are involved with this activity:

- Developer
- Project Leader
- Technical Leader

3.3.10.13.6 Used Tools

This activity uses the following tools:

- Integrated Development Environment
- J2ME
- Real Mobile Device

3.3.10.14 Activity: Identify Scenarios

3.3.10.14.1 Description

The developers go through the customer requirements understanding the requirements, clarifying ambiguities and identifying possible scenarios. A scenario is represented as a UML use case diagram. A scenario has roles and use cases. The scenarios are collected, split, and distributed among developers.

3.3.10.14.2 Input Criteria

1. The collected requirements from the customer were identified clearly as a new development from scratch or a modification of an existing product.

3.3.10.14.3 Exit Criteria

1. A first set of possible scenarios modelled as uml case diagrams.

3.3.10.14.4 Product Flow

This activity consumes the following artifacts:

- Requests From Customer

This activity does not modify any artifacts.
This activity produces the following artifacts:
- Possible Scenarios

3.3.10.14.5 Involved Roles

The following roles are involved with this activity:
- Developer
- Project Leader
- Technical Leader

3.3.10.14.6 Used Tools

This activity uses the following tools:
- UML Editor

3.3.10.15 Activity: Specify Initial Requirements

3.3.10.15.1 Description

The scenarios are specified for the first time as requirements in the SRS. Each developer is responsible for specifying a set of given scenarios.

3.3.10.15.2 Input Criteria

1. A first set of possible scenarios modelled as uml case diagrams.

3.3.10.15.3 Exit Criteria

1. The first draft of the requirements document is provided.

3.3.10.15.4 Product Flow

This activity consumes the following artifacts:
- Possible Scenarios

This activity does not modify any artifacts.
This activity produces the following artifacts:
- SRS

3.3.10.15.5 Involved Roles

The following roles are involved with this activity:
- Developer
- Project Leader
- Technical leader

3.3.10.15.6 Used Tools

This activity uses the following tools:
- Text Editor

3.3.11 Activity: *Feasibility Study*

3.3.11.1 Description

Doubts concerning new technologies are resolved.
The feasibility study is often carried out on the following limitations
1. Heterogeneous clients

2. Network issues
3. Internal and external connection requirements

3.3.11.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.3.11.3 Effort

1. Time spent studying the feasibility and developing guidelines that solve methodological and technological obstacles (Hours).

3.3.11.4 Input Criteria

1. The first draft of the requirements document.

3.3.11.5 Exit Criteria

1. A complete feasibility study that comprises guidelines, which solve methodological and technological obstacles.

3.3.11.6 Product Flow

This activity does not consume any artifacts.
This activity modifies the following artifacts:

- SRS
- Feasibility Study

This activity does not produce any artifacts.

3.3.11.7 Activities

The following are the subactivities of this activity:

- Search Possible Solutions
- Test Possible Solutions

3.3.11.8 Product Flow Refinement

The following graph(s) depict the product flow refinement for this activity.

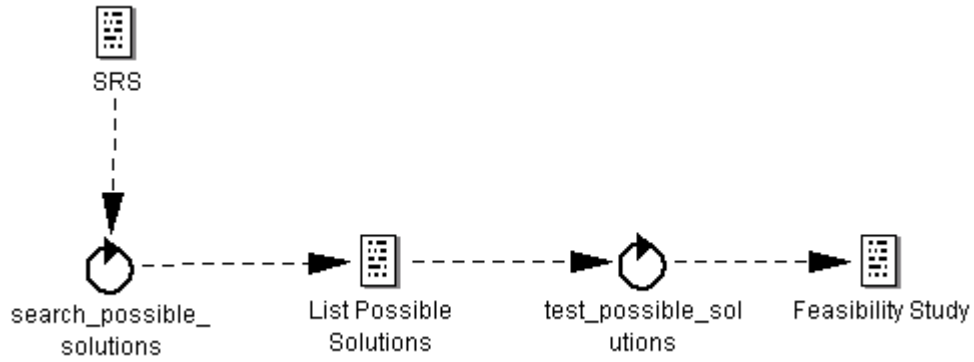

Figure 16. Product flow feasibility study phase.

3.3.11.9 Process View: Roles

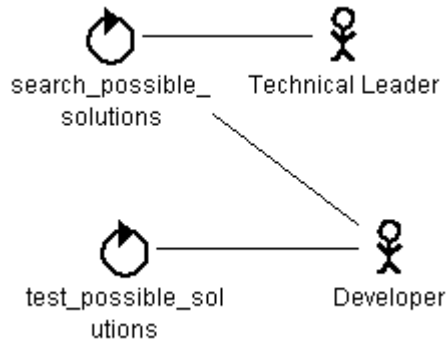

Figure 17. Process view: feasibility study roles.

3.3.11.10 Process View: Tools

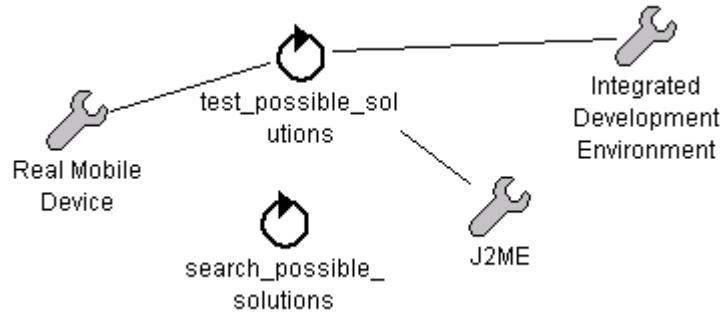

Figure 18. Process view: feasibility study tools.

3.3.11.11 Involved Roles

The following roles are involved with this activity:

- Developer
- Technical Leader

3.3.11.12 Used Tools

This activity uses the following tools:

- Integrated Development Environment
- J2EE
- J2ME
- Real Mobile Device

3.3.11.13 Activity: Search Possible Solutions

3.3.11.13.1 Description

Search in the Intranet or Internet

3.3.11.13.2 Product Flow

This activity consumes the following artifacts:

- SRS

This activity does not modify any artifacts.
This activity produces the following artifacts:
- List Possible Solutions

3.3.11.13.3 Involved Roles

The following roles are involved with this activity:
- Developer
- Technical leader

3.3.11.13.4 Used Tools

This activity uses no tools.

3.3.11.14 Activity: Test Possible Solutions

3.3.11.14.1 Product Flow

This activity consumes the following artifacts:
- List Possible Solutions

This activity does not modify any artifacts.
This activity produces the following artifacts:
- Feasibility Study

3.3.11.14.2 Involved Roles

The following roles are involved with this activity:
- Developer

3.3.11.14.3 Used Tools

This activity uses the following tools:
- Integrated Development Environment
- J2EE
- J2ME
- Real Mobile Device

3.3.12 Activity: *Analyze Requirements*

3.3.12.1 Description

1. Refine SRS draft and produce stable and complete version.
The SRS defines:
1. The functions the product has to perform.
2. Use case descriptions for each function.
3. Inspect the SRS draft.
3.1. Ensure the specification is clear and unambiguous.
3.2. Resolve any questions.
4. The SRS is approved/rejected by the customer

3.3.12.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.3.12.3 Effort

1. Time spent specifying requirements. (Hours)

3.3.12.4 Input Criteria

1. SRS draft approved by the customer.
2. The results of the feasibility study.
3. A defect from a previous version was identified and documented in the defect correction plan. The defect was traced back to a specific requirement. The requirement must be reworked.

3.3.12.5 Exit Criteria

1. A completed requirements specification.
2. The requirements specification must be approved by the customer

3.3.12.6 Product Flow

This activity consumes the following artifacts:
- Feasibility Study
- Defect Log

This activity modifies the following artifacts:
- SRS

This activity does not produce any artifacts.

3.3.12.7 Involved Roles

The following roles are involved with this activity:
- Developer
- Project Leader
- Technical Leader

3.3.12.8 Used Tools

This activity uses the following tools:
- Text Editor

3.4 *ONLINE-ENTERTAINMENT CLIENT / DESIGN PHASE*

3.4.1 Description

1. A high - level or conceptual design is produced
2. A low - level or concrete design is produced
3. An inspection of the design document is executed

3.4.2 Calendar Time

1. Days taken to complete the activities of this phase. (Begin date - End date)

3.4.3 Effort

1. Time spent in the set of activities for this phase. (Hours)

3.4.4 Input Criteria

1. The requirements document approved by the customer.
2. The results of the feasibility study.
3. A defect from a previous version was identified and documented in the defect correction plan. The defect was traced back to the design artifact. The design must be reworked

3.4.5 Exit Criteria

1. A complete design of the logical structure, the components and its interfaces.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets).

3.4.6 Product Flow

This activity consumes the following artifacts:

- Feasibility Study

This activity modifies the following artifacts:

- Design Document

This activity produces the following artifacts:

- Defect Log

3.4.7 Activities

The following are the activities of this phase:

- System Design
- Design
- Design Review

The following graph depicts the product flow for this phase:

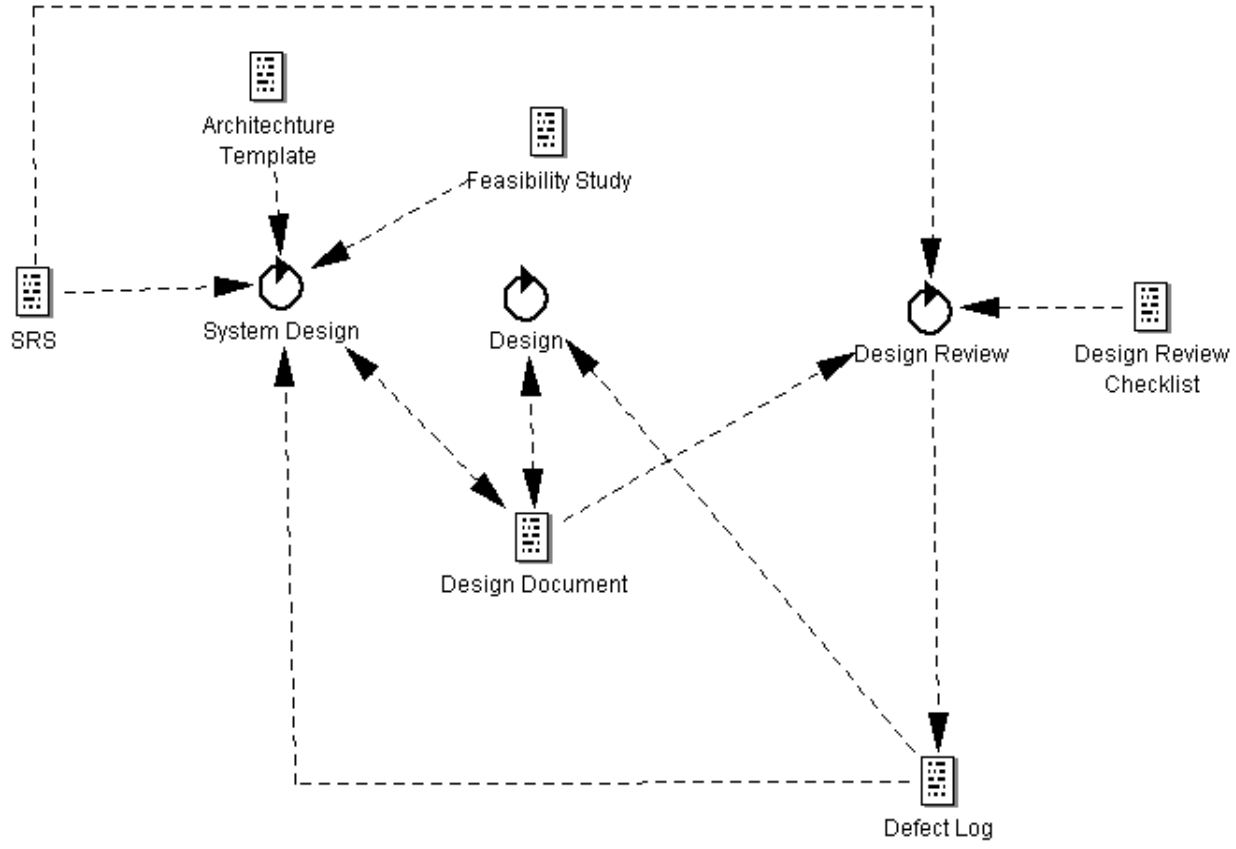

Figure 19. Product flow design phase.

3.4.8 Process View: Roles

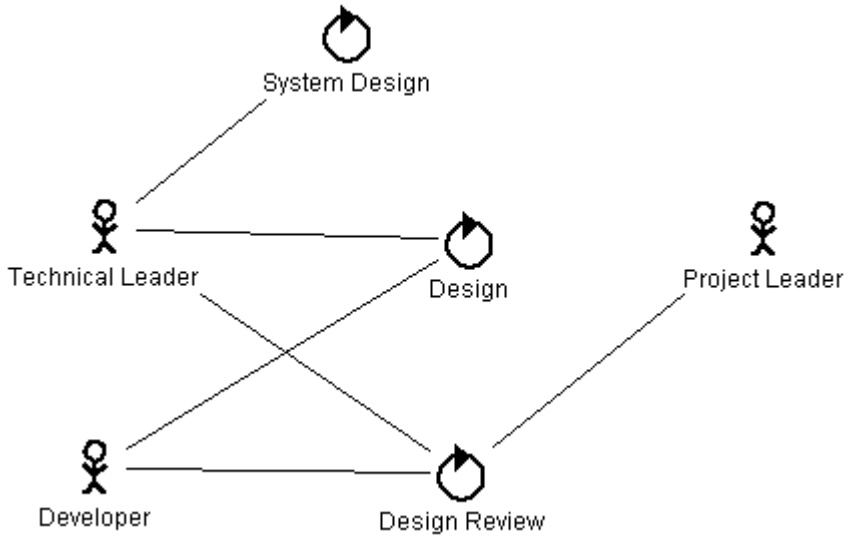

Figure 20. Process view: design phase roles.

3.4.9 Process View: Tools

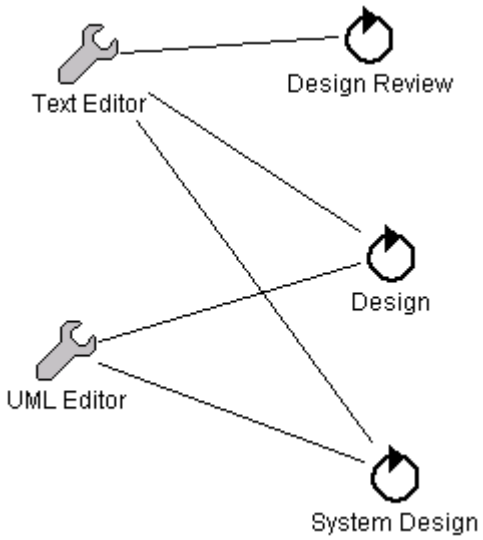

Figure 21. Process view: design phase tools.

3.4.10 Activity: *System Design*

3.4.10.1 Description

Using WISA:
1. Identify new end-user services that belong to the end-user domain category of wireless services depicted in WISA. The category gives an initial list of characteristics, functional and quality requirements.

2. Order quality requirements: the qualities important in achieving the real quality of the end-user service are on top of that priority list.
3. Select the styles and patterns to end-user services that exploit WISA/RA and its basic services according to the identified quality attributes.
4. Use the conceptual structure from WISA/RA to specify the conceptual structure of the new service. The conceptual structure assists in identifying the basic components of the WISA knowledge base that could be exploited in the service development.
5. The identified basic services and components of the WISA knowledge can be reused as such in the structural view of the architecture of the new services. These services and components are defined as COTS, MOTS or OCM components in the development view.
6. Identify the components from WISA and the components that will be developed from scratch
7. The functionality and interfaces for each component are also defined.
Not using WISA:
1. Produce the design document draft.
1.1 Define the product structure
1.2 Name the product components
1.3 Allocate the use cases to these components

3.4.10.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.4.10.3 Effort

1. Time spent designing the product logical structure. (Hours)

3.4.10.4 Input Criteria

1. The requirements document approved by the customer.
2. The results of the feasibility study.
3. A defect was identified and documented in the inspection log template. The design must be reworked.
4. A defect from a previous version was identified and documented in the defect correction plan. The defect was traced back to the design artifact. The design must be reworked.

3.4.10.5 Exit Criteria

1. A complete design of the product logical structure.
2. Calendar time, effort, and produced artifacts size data (Data collection sheets).

3.4.10.6 Product Flow

This activity consumes the following artifacts:

- SRS
- Feasibility Study
- Architecture Template
- Defect Log
-

This activity modifies the following artifacts:

- Design Document

This activity does not produce any artifacts.

3.4.10.7 Involved Roles

The following roles are involved with this activity:

- Technical Leader

3.4.10.8 Used Tools

This activity uses the following tools:

- UML Editor
- Text Editor

3.4.11 Activity: *Design*

3.4.11.1 Description

1. Define the concrete architecture
1.1 Define the structural view.
1.1.1 Define the information classes
1.1.2. Define the associations between classes
1.1.3 Define essential computational-oriented aspects (Create inter component and intra component diagrams)
1.2 Define the concrete behavioural view
1.2.1 Create inter component sequence diagrams
1.2.2 Create intra component sequence diagrams
1.3 Define the concrete deployment view
1.3.1 Define the deployment diagram
1.4 Define the concrete development view
1.4.1 Create a table with the interfaces of components
1.4.2 Describe the technology layers

3.4.11.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.4.11.3 Effort

1. Time spent designing the logical structure, the components and its interfaces. (Hours)

3.4.11.4 Input Criteria

1. A complete design of the product logical structure
2. Guidelines for designing
3. A defect was identified and documented in the inspection log template. The design must be reworked.
4. A defect from a previous version was identified and documented in the defect correction plan. The defect was traced back to the design artifact. The design must be reworked.

3.4.11.5 Exit Criteria

1. A complete design of the logical structure, the components and its interfaces.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets).

3.4.11.6 Product Flow

This activity consumes the following artifacts:

- Defect log

This activity modifies the following artifacts:

- Design Document

This activity does not produce any artifacts.

3.4.11.7 Involved Roles

The following roles are involved with this activity:
- Developer
- Technical Leader

3.4.11.8 Used Tools

This activity uses the following tools:
- UML Editor
- Text Editor

3.4.12 Activity: *Design_Review*

3.4.12.1 Description

The steps are taken from the script INS of the book TSPi (Introduction to the Team Software Process).
1. A moderator and reviewers are selected for the inspection.
2. The moderator makes sure that the product is ready for inspection.
3. The moderator describes the inspection process
3.1 The creator presents briefly the product to the reviewers
3.2 The reviewers select criteria for the inspection. Criteria can be: Operation, recovery, maintenance, security, installation, size, performance, and others (It would be interesting to see criteria that apply for Wireless Internet Services development also)
4. The moderator sets date and time of the inspection
5. The reviewers separately make detailed product reviews
6. The reviewers record their preparation time for the inspection meeting as part of the activity design review.
7. The moderator opens the inspection meeting. If any reviewers are not prepared reschedules the meeting
8. The moderator steps through the design and has the reviewers describe every problem found. The moderator registers the problem in the defect log.
9. The inspection team decides
9.1 Whether a re-inspection is warranted, who should do it, and when
9.2 How to verify the defect corrections
10. The creator
10.1 Makes repairs and update the product documentation
10.2 Calls for a new inspection.

3.4.12.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.4.12.3 Effort

1. Time spent reviewing the design of the logical structure, the components and its interfaces. (Hours)

3.4.12.4 Defects

Defects found during this process. Each defect has the following attributes:
- Defect type:
10. Documentation; 20. Syntax; 30. Build, Package;40. Assignment;50. Interface;60. Checking;70. Data;80. Function;90. System;100. Environment;110. Others
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.

- Fix time
- Severity: High; Medium; Low; Very low
- Comments

3.4.12.5 Input Criteria

1. A complete design of the logical structure, the components and its interfaces.
2. A completed requirements specification.
3. Design review checklist.

3.4.12.6 Exit Criteria

1. The completed defect log with the results of the activity.
2. Modified design.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets).

3.4.12.7 Product Flow

This activity consumes the following artifacts:
- SRS
- Design Document
- Design Review Checklist

This activity does not modify any artifacts.
This activity modifies the following artifacts:
- Defect Log

3.4.12.8 Involved Roles

The following roles are involved with this activity:
- Developer
- Project Leader
- Technical Leader

3.4.12.9 Used Tools

This activity uses the following tools:
- Text Editor

3.5 *ONLINE-ENTERTAINMENT CLIENT / CODING PHASE*

3.5.1 Description

1. The code is produced and tested
2. The code is integrated.
3. The code is released.

3.5.2 Calendar Time

1. Days taken to complete the activities of this phase. (Begin date - End date)

3.5.3 Effort

1. Time spent in the set of activities for this phase. (Hours)

3.5.4 Input Criteria

1. A complete design document.
2. Feasibility study result guidelines
3. A defect from a previous version was identified and documented in the defect correction plan. The defect was traced back to the code artifact. The code must be reworked

3.5.5 Exit Criteria

1. A released and integrated product.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

3.5.6 Product Flow

This activity consumes the following artifacts:
This activity modifies the following artifacts:

- SRS
- Design Document

This activity produces the following artifacts:

- Code Released

3.5.7 Activities

The following are the activities of this phase:

- Code
- Unit Test
- Integrate Code
- Release Code

The following graph depicts the product flow for this phase:

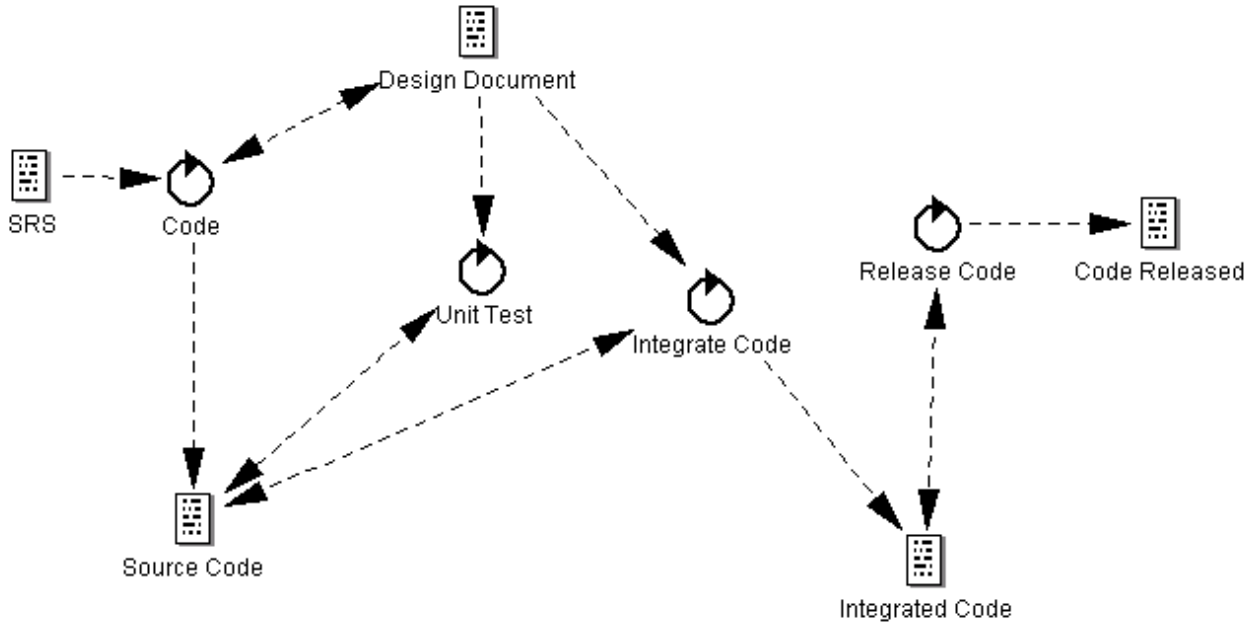

Figure 22. Product flow coding phase.

3.5.8 Process View: Roles

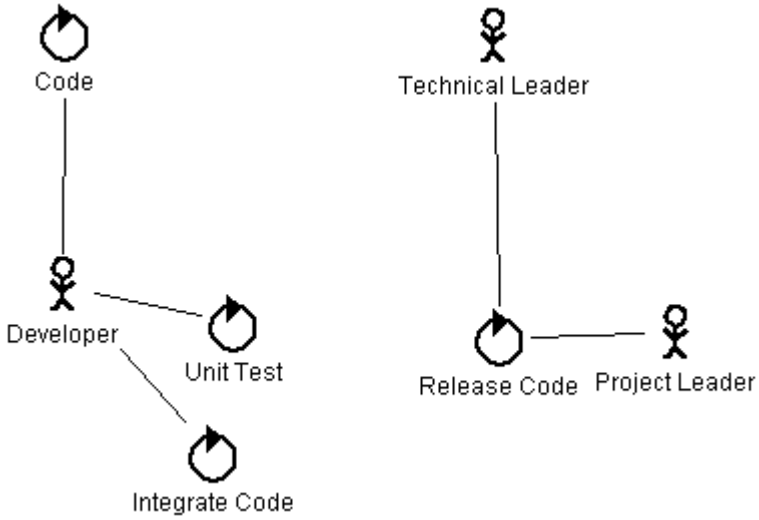

Figure 23. Process view: coding phase roles.

3.5.9 Process View: Tools

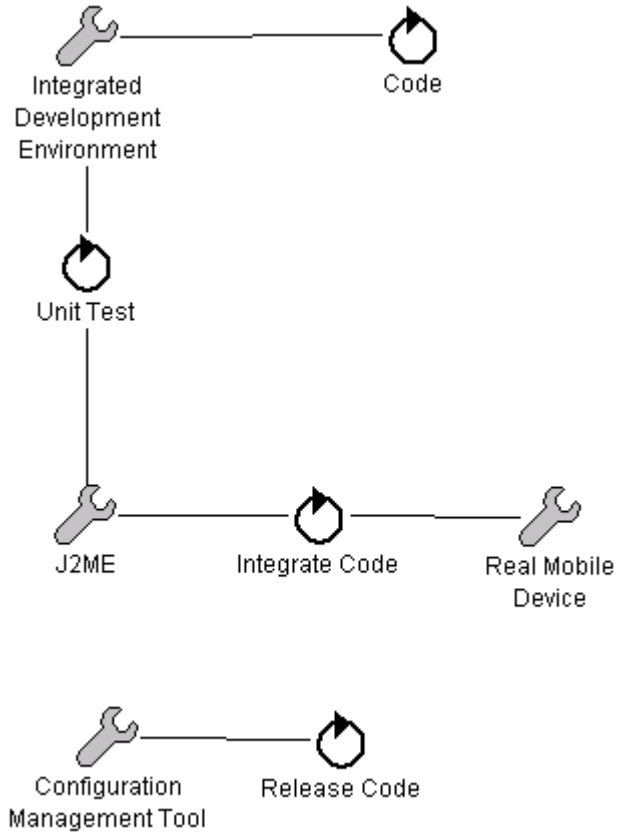

Figure 24. Process view: coding phase tools.

3.5.10 Activity: *Code*

3.5.10.1 Description

1. Define and plan the implementation tasks
2. Produce the component source
2.1 Do a code review using a personal checklist
2.2 Compile and fix the code until it compiles without error

3.5.10.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.5.10.3 Effort

1. Time spent coding. (Hours)

3.5.10.4 Input Criteria

1. A complete design of the logical structure, the components and its interfaces.
2. Feasibility study result guidelines.
3. A defect from a previous version was identified and documented in the defect correction plan. The defect was traced back to the code artifact. The code must be reworked.

3.5.10.5 Exit Criteria

1. Code that complies with the coding standards.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

3.5.10.6 Product Flow

This activity consumes the following artifacts:
- SRS

This activity modifies the following artifacts:
- Design Document

This activity produces the following artifacts:
- Source Code

3.5.10.7 Involved Roles

The following roles are involved with this activity:
- Developer

3.5.10.8 Used Tools

This activity uses the following tools:
- Integrated Development Environment

3.5.11 Activity: Unit Test

3.5.11.1 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.5.11.2 Defects

Defects found during this process. Each defect has the following attributes:
- Defect type: 10. Documentation; 20. Syntax; 30. Build, Package;40. Assignment;50. Interface;60. Checking;70. Data;80. Function;90. System;100. Environment;110. Others
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.
- Fix time
- Severity: High; Medium; Low; Very low
- Comments

3.5.11.3 Description

1. Test the units of code (modules, functions, methods) until tests run without error.
2. Fix all defects found
3. Record defects in the defect log if they are supposed to be injected in the requirements, or design phases.

3.5.11.4 Effort

1. Time spent reviewing the code. (modules, functions, methods) . (Hours)

3.5.11.5 Exit Criteria

1. Units of code (modules, functions, methods) without errors.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

3.5.11.6 Input Criteria

1. Code that complies with the coding standards.
2. A complete design of the logical structure, the components and its interfaces.

3.5.11.7 Product Flow

This activity consumes the following artifacts:
- Design Document

This activity modifies the following artifacts:
- Source Code

This activity does not produce any artifacts.

3.5.11.8 Involved Roles

The following roles are involved with this activity:
- Developer

3.5.11.9 Used Tools

This activity uses the following tools:
- J2ME
- Integrated Development Environment

3.5.12 Activity: *Integrate Code*

3.5.12.1 Description

1. The code for client and for server is integrated (for WISE pilot).
In a 'normal' project (i.e., if the application is large) this activity means that different modules are integrated.

1.1. Verify that all needed parts are on hand.
1.2 Build the product.
2. Execute the integration test plan
2.1 When defects are found it should be determined whether testing should continue
2.2 Every defect found is recorded in the defect log and review to determine:
Where similar defects may remain in the product.
How and when to find and fix these defects.

3.5.12.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.5.12.3 Defects

Defects found during this process. Each defect has the following attributes:
- Defect type: 10. Documentation; 20. Syntax; 30. Build, Package;40. Assignment;50. Interface;60. Checking;70. Data;80. Function;90. System;100. Environment;110. Others
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.
- Fix time
- Severity: High; Medium; Low; Very low
- Comments

3.5.12.4 Effort

1. Time spent integrating the code. (Hours)

3.5.12.5 Input Criteria

1. Units of code (modules, functions, methods) without errors.
2. Feasibility study results.
3. A complete design of the logical structure, the components and its interfaces.
4. A defect from a previous version was identified and documented in the defect correction plan. The defect was traced back to the integrated code artifact. The integrated code must be reworked.

3.5.12.6 Exit Criteria

1. An integrated and tested product
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

3.5.12.7 Product Flow

This activity consumes the following artifacts:
- Design Document

This activity modifies the following artifacts:
- Source Code

This activity produces the following artifacts:
- Integrated Code

3.5.12.8 Involved Roles

The following roles are involved with this activity:
- Developer

3.5.12.9 Used Tools

This activity uses the following tools:

- J2ME
- Real Mobile Device

3.5.13 Activity: *Release Code*

3.5.13.1 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.5.14 Description

1. The code is satisfactorily implemented, and integrated
2. A version number is associated to the code.
3. The objectives and features of the code are released.
4. The code is entered in the configuration management system

3.5.14.1 Effort

1. Time spent releasing the code. (Hours)

3.5.14.2 Exit Criteria

1. A released and integrated product.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

3.5.14.3 Input Criteria

An integrated and tested product
Defect correction plan

3.5.14.4 Product Flow

This activity consumes the following artifacts:
This activity modifies the following artifacts:

- Integrated Code

This activity produces the following artifacts:

- Code Released

3.5.14.5 Involved Roles

The following roles are involved with this activity:

- Project Leader
- Technical Leader

3.5.14.6 Used Tools

This activity uses the following tool:

- Configuration Management Tool

3.6 *ONLINE-ENTERTAINMENT CLIENT / TESTING PHASE*

3.6.1 Description

1. Tests are planned
2. The framework or environment (i.e., software, hardware) is settled.
3. The system test is executed.
4. The system is released and tested by the end users
5. The results from end users testing is analysed and a new correction plan is produced.

3.6.2 Calendar Time

1. Days taken to complete the activities of this phase. (Begin date - End date)

3.6.3 Effort

1. Time spent in the set of activities for this phase. (Hours)

3.6.4 Input Criteria

1. A released and integrated product.
2. The test report template format.

3.6.5 Exit Criteria

1. A complete report including the results from all tests.
2. A released, integrated, and user tested product.
3. A complete plan for fixing the defects in the actual or later version of the product.

3.6.6 Product Flow

This activity consumes the following artifacts:

- SRS
- Design Document

This activity modifies the following artifacts:

- Code Released
- Test Report

This activity produces the following artifacts:

- Product Tested

3.6.7 Activities

The following are the activities of this phase:

- Plan Tests
- Build Test Framework
- Test System
- Acceptance Test
- Analyze Defect
- Test Usability

3.6.8 Product Flow Refinement

The following graph depicts the product flow for this phase:

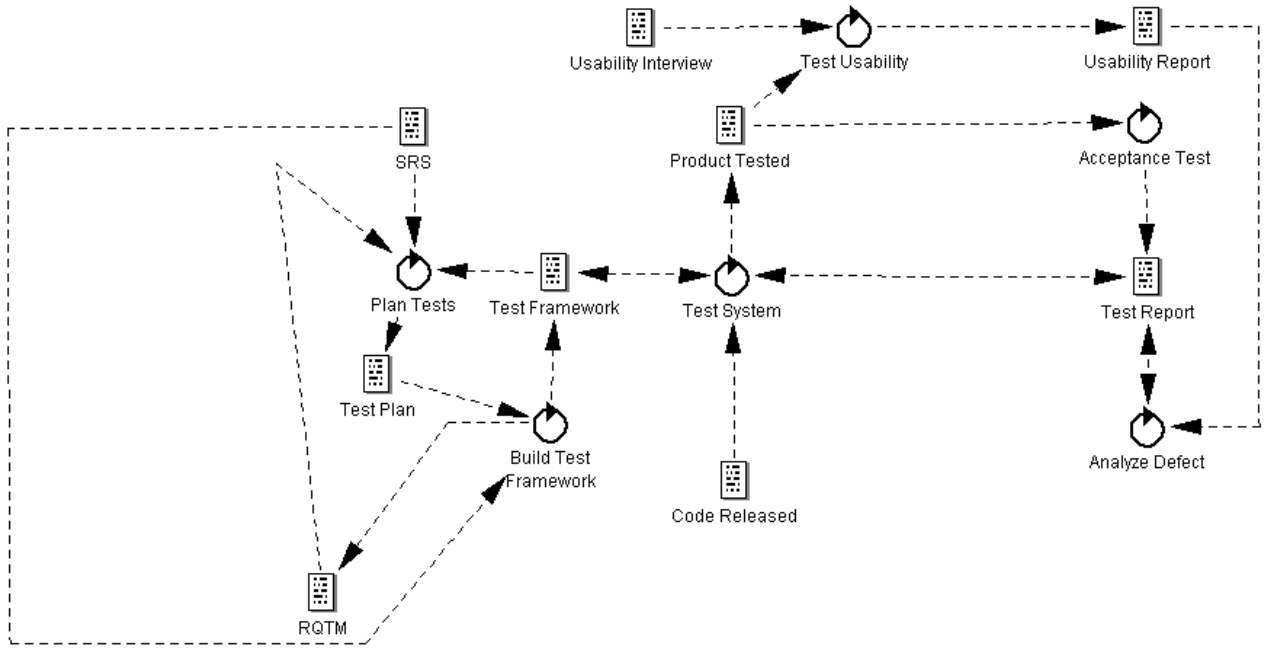

Figure 25. Product flow feature test phase.

3.6.9 Process View: Roles

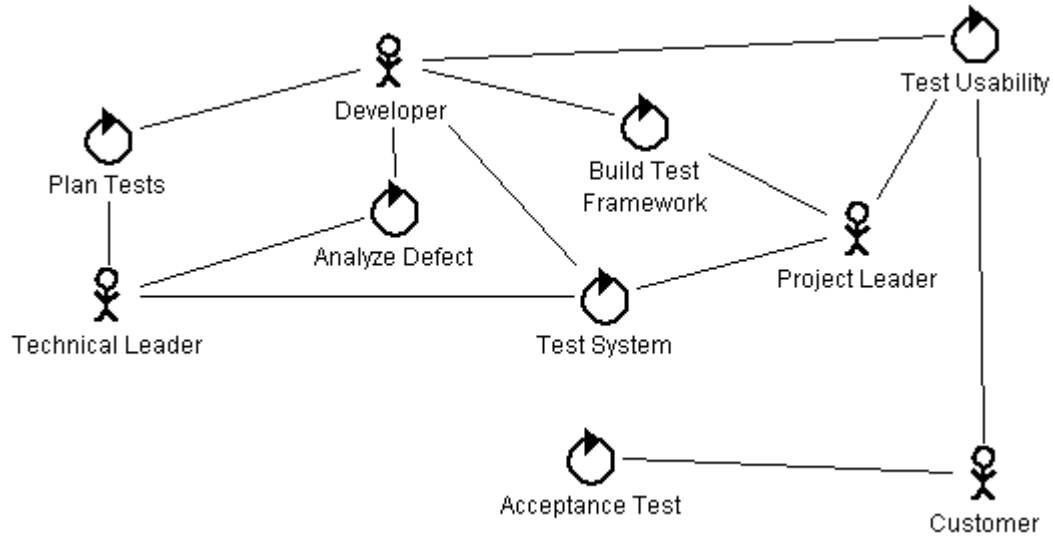

Figure 26. Process view: testing phase roles.

3.6.10 Process Views: Tools

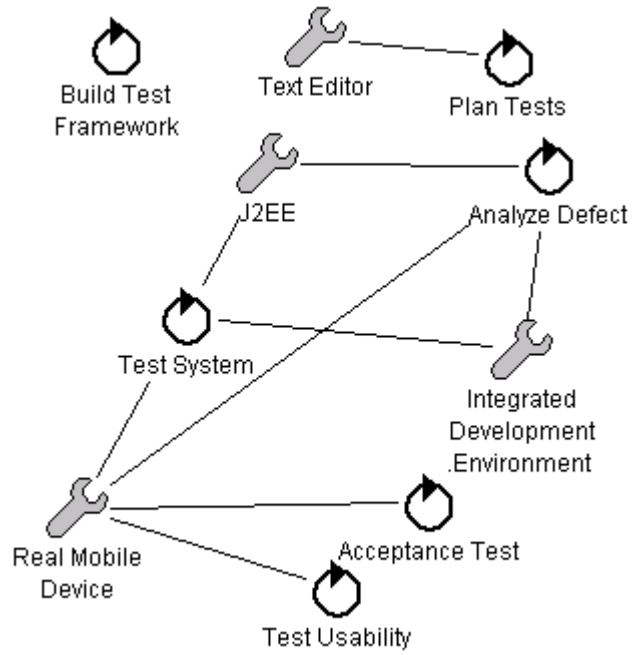

Figure 27. Process view: design phase roles.

3.6.11 Activity: *Plan Tests*

3.6.11.1 Description

Based on the customer requests and the SRS, testing is planned. Re-planning the test is possible when new issues appear after setting up the test framework of mapping the test cases to the requirements in the RQTM.

1. Estimate the sequence the test plan has to be developed
2. Estimate the effort to accomplished the test plan
3. Identify the resources for executing the test plan (configuration)

3.6.11.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.6.11.3 Effort

1. Time spent planning tests with the sequence of test cases and a description of the needed environment to test the product (Hardware and software) (Hours).

3.6.11.4 Input Criteria

1. A completed requirements specification.
2. The set of requirements mapped to the test cases in the RQTM.
3. A configured test framework.

3.6.11.5 Exit Criteria

1. A complete test plan with the sequence of test cases and a description of the needed environment to test the product (Hardware and software).

3.6.11.6 Product Flow

This activity consumes the following artifacts:
- SRS
- Test Framework
- RQTM

This activity does not modify any artifacts.
This activity produces the following artifacts:
- Test Plan

3.6.11.7 Involved Roles

The following roles are involved with this activity:
- Developer
- Technical Leader

3.6.11.8 Used Tools

This activity uses the following tools:
- Text Editor

3.6.12 Activity: *Build Test Framework*

3.6.12.1 Description

1. Based upon the test documents (test plan, test procedure document, and test case document) a test framework (i.e., an environment in which the tests are to be performed) is built.
2. The RQTM is produced. All test cases are traced back into the RQTM.

3.6.12.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.6.12.3 Effort

1. Time spent setting the hardware and software required to test the product (Hours).

3.6.12.4 Input Criteria

1. A complete test plan with the sequence of test cases and a description of the needed environment to test the product (Hardware and software).
2. A complete requirements specification.

3.6.12.5 Exit Criteria

1. The hardware and software required to test the product is settled.
2. The set of requirements mapped to the test cases in the RQTM.
3. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

3.6.12.6 Product Flow

This activity consumes the following artifacts:
- SRS
- Test Plan

This activity does not modify any artifacts.
This activity produces the following artifacts:
- Test Framework

- RQTM

3.6.12.7 Involved Roles

The following roles are involved with this activity:
- Developer
- Project Leader

3.6.13 Activity: *Test System*

3.6.13.1 Description

1. Test the product for normal and stress conditions
1.1 The system (integrated on cell phone) is tested.
In WISE: Feature tests/system tests are performed. In particular this tests, the connection between the server and the client.
2. Test the product for installation, conversion and recovery
3. Apply regression test to the system
4. Record all the test activities

3.6.13.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.6.13.3 Defects

Defects found during this process. Each defect has the following attributes:
- Defect type: 10. Documentation; 20. Syntax; 30. Build, Package;40. Assignment;50. Interface;60. Checking;70. Data;80. Function;90. System;100. Environment;110. Others
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.
- Fix time
- Severity: High; Medium; Low; Very low
- Comments

3.6.13.4 Effort

Time spent testing the system (Hours).
Time spent for recording the performed test activities (cases). (Hours)

3.6.13.5 Input Criteria

1. A released and integrated product.
2. The hardware and software required to test the product is settled.

3.6.13.6 Exit Criteria

1. A released and system tested product.
2. Report with record from the performed test activities (cases).
3. Calendar time, effort, and produced artifacts size data. (Data collection sheets).

3.6.13.7 Product Flow

This activity consumes the following artifacts:
- Code Released

This activity modifies the following artifacts:
- Source Code
- Test Framework
- Test Report

This activity produces the following artifacts:
- Product Tested

3.6.13.8 Involved Roles

The following roles are involved with this activity:
- Developer
- project Leader
- technical Leader

3.6.13.9 Used Tools

This activity uses the following tools:
- J2EE
- Integrated Development Environment
- Real Mobile Device

3.6.14 Activity: *Acceptance_Test*

3.6.14.1 Description

Cell phone with new functionality is tested by a friendly customer.
In WISE: role of friendly customer may be assumed by WISE team. Not yet clear how to perform in WISE.

3.6.14.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.6.14.3 Defects

Defects found during this process. Each defect has the following attributes:
- Defect type: 10. Documentation; 20. Syntax; 30. Build, Package;40. Assignment;50. Interface;60. Checking;70. Data;80. Function;90. System;100. Environment;110. Others
- Injecting process
- Detecting process
- Involved product
- Defect description
- Defect qualifier: Incorrect; Missing; Extraneous.
- Fix time
- Severity: High; Medium; Low; Very low
- Comments

3.6.14.4 Effort

1. Time spent performing the acceptance test (Hours).

3.6.14.5 Input Criteria

1. A released and system tested product.

3.6.14.6 Exit Criteria

1. Report with record from the performed test activities (cases).

3.6.14.7 Product Flow

This activity consumes the following artifacts:
- Product Tested

This activity does not modify any artifacts.
This activity produces the following artifacts:
- Test Report

3.6.14.8 Involved Roles

The following roles are involved with this activity:
- Customer

3.6.14.9 Used Tools

This activity uses the following tools:
- Integrated Development Environment
- real Mobile Device

3.6.15 Activity: *Analyze Defect*

3.6.15.1 Description

This activity is performing in parallel with the acceptance tests, and the usability test.
The report is:
1. If a defect, problem, or suggestion found is considered to be serious, a lists of tasks to solve it is created and the responsibles assigned. Replanning concerning the delivery of the product to the market must be made. If the defect is considered to be minor, then a requirement for the new version of the product is generated, therefore, no delays on scheduled delivery are assumed.

3.6.15.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.6.15.3 Effort

1. Time spent planning defects correction in the actual or latest version of the product. (Hours).

3.6.15.4 Input Criteria

1. Report with record from the performed test activities (cases).

3.6.15.5 Exit Criteria

1. A complete plan for fixing the defects in the actual or later version of the product.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

3.6.15.6 Product Flow

This activity consumes the following artifacts:
- Usability Report

This activity modifies the following artifacts:
- Test Report

This activity does not produce any artifacts.

3.6.15.7 Involved Roles

The following roles are involved with this activity:
- Developer
- Technical Leader

3.6.15.8 Used Tools

This activity uses the following tools:
- Integrated Development Environment
- J2EE
- Real Mobile Device

3.6.16 Activity: *Test Usability*

3.6.16.1 Description

This technique is based in Nielsen's [1] in which experts guided by a set of usability principles known as heuristics evaluate whether user-interface elements such as dialog boxes, menus, navigation structure, online help, etc conform to the principles.
5 evaluators are recommended.
1. The briefing session: Experts are told what to do. A prepared script is useful as a guide and to ensure each person receives the same briefing.
2. The evaluation period: Each expert typically spends 1-2 hours independently inspecting the product using the heuristics for guidance. The experts need to take at least two passes through the interface. The first pass gives a feel of the flow interaction and the product scope, the second allows the evaluator to focus on specific interface elements in the context of the whole products. Pre-planned tasks may be helpful.
2.1. While working on the interfaces the evaluator must record the problems.
3. The debriefing session in which experts come together to discuss their findings and to prioritize the problems they found and suggestions
[1] Nielsen, J., Mack, R.L.: Usability Inspection Methods / John Wiley & Sons, Inc; (1994).

3.6.16.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

3.6.16.3 Effort

1. Time spent planning defects correction in the actual or latest version of the product. (Hours).

3.6.16.4 Input Criteria

1. A released and system tested product.
2. A usability checklist

3.6.16.5 Exit Criteria

1. A completed usability report

3.6.16.6 Product Flow

This activity consumes the following artifacts:
- Product Tested
- Usability Interview

This activity does not modify any artifacts.
This activity produces the following artifacts:
- Usability Report

3.6.16.7 Involved Roles

The following roles are involved with this activity:
- Customer
- Developer
- Technical Leader

3.6.16.8 Used Tools

This activity uses the following tools:
- Real Mobile Device

3.7 ONLINE-ENTERTAINMENT CLIENT/ARTIFACTS

3.7.1 Artifact: *Requests From Customer*

3.7.1.1 Description

The requests from customer can be produced in two different situations: new service or change to a previous service
New service: In this case the document is usually a description of a new service designed for one specific mobile device. The document is elaborated by the marketing division and describes a wished user interface and the functionality of the service.
Change to a previous service: Bugs, or suggestions for improvement found in previous releases of a service can be documented as customer requirements by the customers.

3.7.1.2 Product Flow

This artifact is not produced by any activity.
This artifact is modified by the following activities.
- Gather Requirements
- Analyze User Interface Feasibility Requirements

This artifact is used by the following activities:
- Requirements Phase
- Identify Scenarios

3.7.2 Artifact: *SRS*

3.7.2.1 Description

System Requirements Specification - the SRS should contain system requirements and a link to the RQTM.
Sample of the contents:
0. Table of contents
1. Introduction

2. Service architecture: It provides the general framework on which the service is based, and general constraints to be considered.
3. Service functional requirements: describe requirements at the level of the entire service.
4. Client side functional requirements: The use cases of the client side of the service are listed. For each use case, associated functional requirements are listed.
5. Server side functional requirements: The use cases of the server side of the service are listed. For each use case, associated functional requirements are listed.
6. Performance and usability requirements. For example: Mobile device memory limitation.
. Examples of usability and performance requirements are:
- Requirement 001: Midlets at runtime shall not require more than 662Kb. For mobile devices, runtime memory availability may be even lower.
- Requirement 002: A dedicated thread should be responsible for network access and management of the user, so the user can continue interacting with the device at all times.

3.7.2.2 Product Flow

This artifact is produced by the following activities:
- Gather Requirements
- Specify Initial Requirements

This artifact is modified by the following activities.
- Requirements Phase
- Feasibility Study
- Analyze Requirements
- Coding Phase

This artifact is used by the following activities:
- System Design
- Design Review
- Code
- Testing Phase
- Plan Tests
- Build Test Framework
- Search Possible Solutions

3.7.3 *Artifact*: *List Possible Solutions*

3.7.3.1 Description

List with the recommendations from experts, Internet hits, literature references.

3.7.3.2 Product Flow

This artifact is produced by the following activities:
- Search Possible Solutions

This artifact is not modified by any activity.
This artifact is used by the following activities:
- Test Possible Solutions

3.7.4 *Artifact*: *Possible Scenarios*

3.7.4.1 Description

This artifact consists of a set of possible identified scenarios described as UML use case diagrams.

3.7.4.2 Product Flow

This artifact is produced by the following activities:
- Identify Scenarios

This artifact is not modified by any activity.
This artifact is used by the following activities:
- Specify Initial Requirements

3.7.5 Artifact: *Feasibility Study*

3.7.5.1 Description

Definition of the main problems when developing an application for wireless Internet services and guidelines to solve these problems during the life cycle of development.

This document could help for calibrating requirements between both parts Sonera and Motorola. Important technical issues to address in this document are for example: Is the band with enough? What is the processing speed of the client? How will the client and server communicate?

3.7.5.2 Product Flow

This artifact is produced by the following activities:
- Test Possible Solutions

This artifact is modified by the following activities.
- Requirements Phase
- Feasibility Study

This artifact is used by the following activities:
- Analyze Requirements
- Design phase
- System Design

3.7.6 Artifact: *Architecture Template*

3.7.6.1 Description

The format is taken from the guidelines to describe the architecture depicted at D4a.
The contents of specific architectural documents depend on the needs emerging during the development process. Usually they contain one or more architectural views, selected among those defined by the guidelines. The table of contents of the architectural documents can be organized based on the selected views.

1. Conceptual architecture
1.1 Conceptual structural view
1.1.1 System context (structure of the network)
1.1.2. Domain information models
1.1.3. Functional structure
1.2 Conceptual behavioural view
1.2.1 Collaboration diagram
1.3 Conceptual deployment view
1.3.1 Deployment diagram
1.4 Conceptual development view
1.4.1 Business model
1.4.2 Topology diagram
2. Concrete architecture
2.1 Structural view.

2.1.1 Information classes
2.1.2. Associations between classes
2.1.3 Essential computational-oriented aspects (Create inter component and intra component diagrams)
2.2 Concrete behavioural view
2.2.1 Inter component sequence diagrams
2.2.2 Intra component sequence diagrams
2.3 Concrete deployment view
2.3.1 Deployment diagram
2.4 Concrete development view
2.4.1 Table with the interfaces of components
2.4.2 Technology layers
Other guideline

3.7.6.2 Product Flow

This artifact is not produced by any activity.
This artifact is not modified by any activity.
This artifact is used by the following activities:

- System Design

3.7.7 Artifact: *Defect Log*

3.7.7.1 Description

Log of defects found during the review of the design.

3.7.7.2 Product Flow

This artifact is produced by the following activities:

- Design Phase
- Design Review

This artifact is not modified by any activity.
This artifact is used by the following activities:

- Requirements Phase
- Analyze Requirements
- System Design
- Design

3.7.8 Artifact: *Design Document*

3.7.8.1 Description

Summarizes the detailed design.
It contains the designs of:
The set of classes, which are part of each component, and its attributes, the interaction among classes in order to provided the desired functionality

3.7.8.2 Product Flow

This artifact is not produced by any activity.
This artifact is modified by the following activities.

- Design Phase
- System Design
- Design

- Coding Phase
- Code

This artifact is used by the following activities:

- Design Review
- Unit Test
- Integrate code
- Testing Phase

3.7.9 Artifact: *Design Review Checklist*

3.7.9.1 Description

The checklist is intended for reviewing the concrete architecture. Please go to http://wise.iese.fhg.de/

3.7.9.2 Product Flow

This artifact is not produced by any activity.
This artifact is not modified by any activity.
This artifact is used by the following activities:

- Design Review

3.7.10 Artifact: *Source Code*

3.7.10.1 Description

Lines of code written that conforms to the coding standard

3.7.10.2 Product Flow

This artifact is produced by the following activities:

- Code

This artifact is modified by the following activities.

- Unit Test
- Integrate Code
- Test System

This artifact is not used by any activity.

3.7.11 Artifact: *Integrated Code*

3.7.11.1 Description

Integrated and tested code that conforms to the coding standard

3.7.11.2 Product Flow

This artifact is produced by the following activities:

- Integrate Code

This artifact is modified by the following activities.

- Release Code

This artifact is not used by any activity.

3.7.12 Artifact: *Code Released*

3.7.12.1 Description

Code implemented, integrated, tested, and entered in the configuration management system.

3.7.12.2 Product Flow

This artifact is produced by the following activities:
- Coding Phase
- Release Code

This artifact is modified by the following activities.
- Testing Phase

This artifact is used by the following activities:
- Test System

3.7.13 Artifact: *Test Plan*

3.7.13.1 Description

Describes what tests are to be performed, using what configuration, etc. Based on the requirements defined in the SRS.
The test plan should contain:
1. The objectives for every test
2. The procedures to be performed for every test
3. The results the tests are to produce
4. The supporting materials required for each test.

3.7.13.2 Product Flow

This artifact is produced by the following activities:
- Plan Tests

This artifact is not modified by any activity.
This artifact is used by the following activities:
- Build Test Framework

3.7.14 Artifact: *Test Framework*

3.7.14.1 Description

Framework that allows testing the software integrated on cell phone. The test framework should at least contain a complete environment to test the product. It should consist at least of:
1. Test Procedures document
2. Test Plan
3. Stubs (code that simulates real environment or specific error scenarios)

3.7.14.2 Product Flow

This artifact is produced by the following activities:
- Build Test Framework

This artifact is modified by the following activities.
- Test System

This artifact is used by the following activities:
- Plan Tests

3.7.15 Artifact: *RQTM*

3.7.15.1 Description

The Requirements traceability matrix maintains the traceability among the SRS, and the Test cases.
The traceability matrix should have the following columns:
1. Requirement ID
2. Test case ID

3.7.15.2 Product Flow

This artifact is produced by the following activities:
- Build Test Framework

This artifact is not modified by any activity.
This artifact is used by the following activities:
- Plan Tests

3.7.16 Artifact: *Product Tested*

3.7.16.1 Description

Integrated and tested system ready to be released to the end customers.

3.7.16.2 Product Flow

This artifact is produced by the following activities:
- Testing Phase
- Test System

This artifact is not modified by any activity.
This artifact is used by the following activities:
- Acceptance Test
- Test Usability

3.7.17 Artifact: *Usability Interview*

3.7.17.1 Description

This document contains the questions to be followed by the users in order to evaluate the usability of the application.
The structure is as follows:
- Scope
- Questions for Evaluating Web Sites
- Questions for Online Communities
- Questions for Video Games
- Problems and Suggestions

3.7.17.2 Product Flow

This artifact is not produced by any activity.
This artifact is not modified by any activity.

This artifact is used by the following activities:
- Test Usability

3.7.18 Artifact: *Usability Report*

3.7.18.1 Description

This document contains the summary of the problems found in the usability test, and the suggestions made in order to improve the product with their respective priority.
The structure is:
- SCOPE
- List of problems/suggestions order by priority

3.7.18.2 Product Flow

This artifact is produced by the following activities:
- Test usability

This artifact is not modified by any activity.
This artifact is used by the following activities:
- Analyze Defect

3.7.19 Artifact: *Test Report*

3.7.19.1 Description

The test report should contain:
1. The date the system test was run
2. The name of the person running the system test
3. The test's name and number
4. The product and configuration tested
5. The time each test started
6. The time each test ended
7. The number of defects found
8. The test results

3.7.19.2 Product Flow

This artifact is produced by the following activities:
- Acceptance Test

This artifact is modified by the following activities.
- Testing Phase
- Test System
- Analyze Defect

This artifact is not used by any activity.

3.8 ONLINE-ENTERTAINMENT CLIENT/ROLES

3.8.1 *Role: Customer*

3.8.1.1 Description

This role is assumed by individuals with knowledge and experience on designing and creating new services and products for the mobile devices of the company.

3.8.1.2 Participation

This role participates in the following activities:
- Gather Requirements
- Acceptance Test
- Test Usability

3.8.2 Role: *Developer*

3.8.2.1 Description

This role is assumed by the members of the project with knowledge and experience in programming Wireless Internet Applications.
They are in charge of developing the service across all its lifecycle. They take part in estimating the effort, analyzing requirements, designing the service, implementing and testing it.
They all may share the task of configuring the various environments and manage the configuration. They are assigned on different tasks according to a planning done by the Project Leader and Project Manager. The required skills for this domain are:
- Programming skills / Project support skills
- Knowledge of the wireless environment
- Network programming and configuration, or embedded software experience are preferable

3.8.2.2 Participation

This role participates in the following activities:
- Feasibility Study
- Analyze Requirements
- Design
- Design Review
- Code
- Unit Test
- Integrate Code
- Plan Tests
- Build Test Framework
- Test System
- Analyze Defect
- Analyze User Interface Feasibility Requirements
- Identify Scenarios
- Specify Initial Requirements
- Test Usability
- Search Possible Solutions
- Test Possible Solutions

3.8.3 *Role*: *Project Leader*

3.8.3.1 Description

This role is assumed by the members of the project with knowledge and experience programming Wireless Internet Applications, and experience as managers of software development projects.
He is in charge of the relationships with the client and to elicit the requirements from the client.
He is in charge of managing the project, the project plan. The project plan is usually filled in with the Project Manager.
He is in charge of monitoring and controlling the project.
He is in charge of evaluating the risks and issues and finding out how to solve them.
He is a team member, so in absence of urgent issues he takes part in the development of the project.

3.8.3.2 Participation

This role participates in the following activities:

- Gather Requirements
- Analyze Requirements
- Design Review
- Release Code
- Build Test Framework
- Test System
- Analyze User Interface Feasibility Requirements
- Identify Scenarios
- Specify Initial Requirements
- Test Usability

3.8.4 *Role*: *Project Manager*

3.8.4.1 Description

Manages people and resources. Gets the project and is the general responsible of it.
He is responsible for managing costs and staffing.
He is responsible for the team members working time (he authorizes extra working days and overtime work together with the senior management).
He is responsible in negotiating with the client the costs and the staffing for the project and, in case of grave issues, to re-negotiate them.

3.8.4.2 Participation

This role does not participate in any activity.

3.8.5 Role: *Technical Leader*

3.8.5.1 Description

Leads the technical issues
He is responsible for the design documents. He approves them together with the project leader
He reviews the SRS and give the general guidelines for design.
He knows the technology and clarifies developers technical doubts.
He is capable of following the different tasks in which the developers are involved and giving advices to them.
He masters the situation of the development and reports the risks to the project leader.
In case of big projects more technical leaders can manage different technical areas of the project.
He is a team member, so in absence of urgent issues he takes part in the development of the project.

3.8.5.2 Participation

This role participates in the following activities:

- Gather Requirements
- Feasibility Study
- Analyze Requirements
- System Design
- Design
- Design Review
- Release Code
- Plan Tests
- Test System
- Analyze Defect

- Analyze User Interface Feasibility Requirements
- Identify Scenarios
- Specify Initial Requirements
- Search Possible Solutions

3.9 ONLINE-ENTERTAINMENT CLIENT/TOOLS

3.9.1 *Tool*: *J2EE*

3.9.1.1 Description

J2EE technology and its component based model simplify enterprise development and deployment. The J2EE platform manages the infrastructure and supports the Web services to enable development of secure, robust and interoperable business applications. The J2EE platform is the foundation technology of the Sun ONE platform and Sun's Web services strategy
Taken from http://java.sun.com/j2ee/

3.9.1.2 Usage

This tool is used by the following activities:

- Feasibility Study
- test System
- Analyze Defect
- Test Possible Solutions

3.9.2 *Tool*: *J2ME*

3.9.2.1 Description

The Java 2 Platform Micro Edition, (J2ME) Wireless Toolkit are sets of tools that provide application developers with the emulation environments, documentation and examples needed to develop Java technology applications targeted at CLDC/MIDP compliant mobile phones and entry level PDAs. Two products are available depending on your needs. Both tools are available at no charge to individual applications developers.
For developing applications for MIDP 1.0 devices, use the J2ME Wireless Toolkit version 1.0.4_01.
For developing applications for MIDP 2.0 devices, use the J2ME Wireless Toolkit version 2.0. The 2.0 version also includes built-in support for the Wireless Messaging and Mobile Media APIs.

Taken from http://java.sun.com/j2me/

3.9.2.2 Usage

This tool is used by the following activities:

- Feasibility Study
- Unit Test
- integrate Code
- Analyze User Interface Feasibility Requirements
- Test Possible Solutions

3.9.3 Tool: *UML Editor*

3.9.3.1 Description

Tool for creating, and maintaining documents whose contents are standard UML-Diagrams.

3.9.3.2 Usage

This tool is used by the following activities:

- System Design
- Design
- Identify Scenarios

3.9.4 Tool: *Configuration Management Tool*

3.9.4.1 Usage

This tool is used by the following activities:

- Release Code

3.9.5 Tool: *Integrated Development Environment*

3.9.5.1 Description

An integrated development environment (IDE) is a programming environment that has been packaged as an application program, typically consisting of a code editor, a compiler, a debugger, and a graphical user interface (GUI) builder. The IDE may be a standalone application or may be included as part of one or more existing and compatible applications.
Taken from http://search390.techtarget.com/gDefinition/0,294236,sid10_gci754848,00.html
Important integrated environment for Java are:
Visual Age for Java
JBuilder
JDK
Sun Studio: This environment was improved for using J2ME.

3.9.5.2 Usage

This tool is used by the following activities:

- Feasibility Study
- Code
- Unit Test
- Test System
- Acceptance Test
- Analyze Defect
- Analyze User Interface Feasibility Requirements
- Test Possible Solutions

3.9.6 Tool: *Real Mobile Device*

3.9.6.1 Description

Mobile devices with support for 2G, 3G wireless networks are:
- Desktops/Notebooks
- Tablet-PC
- PDA
- Handheld PC
- PDA- Phones

- Smart - Phones
- Cellular - Phones

3.9.6.2 Usage

This tool is used by the following activities:

- Feasibility Study
- Integrate Code
- Test System
- Acceptance Test
- Analyze Defect
- Analyze User Interface Feasibility Requirements
- Test Usability
- Test Possible Solutions

3.9.7 Tool: *Text Editor*

3.9.7.1 Usage

This tool is used by the following activities:

- Gather Requirements
- Analyze Requirements
- System Design
- Design
- Design Review
- Plan Tests
- Specify Initial Requirements

4. PROCESS MODEL FOR PILOT 2 SERVER: *ONLINE-ENTERTAINMENT*

This model describes the process followed by Pilot 2 Server during the third iteration of the WISE project. The information in the model is based mainly on examples of the processes as they have been performed in the respective environments.

4.1 OVERVIEW OF THE PROCESS MODEL

4.1.1 Activities

The following are the activities present in this process model:

- Exploration Phase
- Planning Phase
- Release Phase
    - Analysis
    - Design
    - Programming
        - Test First
        - Pair Programming
        - Test
        - Continuous Integration
        - Test System

4.1.2 Artifacts

The following are the artifacts in this process model:

- Requirements Document
- Analysisd
- Designd
- Test Code
- Test Cases
- Code
- Integrated Code
- Release

4.1.3 Roles

The following are the roles present in this process model:

- Customer
- Developer
- Project Leader

4.1.4 Tools

The following are the tools present in this process model:

- J2EE
- UML Editor
- Integrated Development Environment
- Text Editor

4.2 ONLINE-ENTERTAINMENT SERVER /PHASE_OVERVIEW

The following are the phases described in this process description:

- Exploration Phase

- Planning Phase
- Release Phase

4.2.1 Product Flow Among Phases

The following graph depicts the product flow between the different phases.

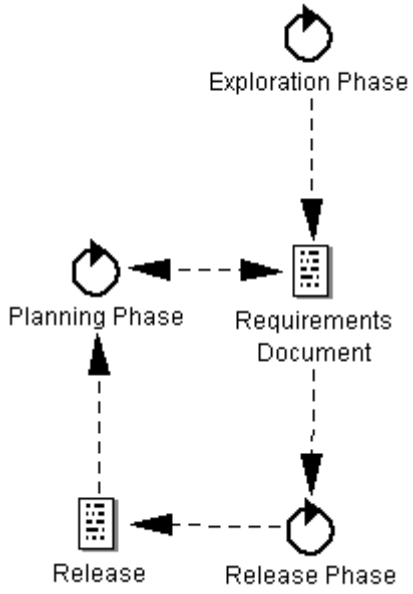

Figure 28. Product flow between phases.

4.2.2 Process View : Roles

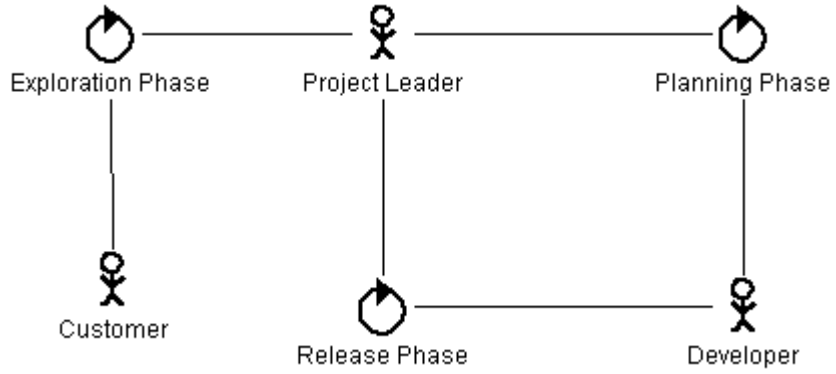

Figure 29. Process view: roles.

4.2.3 Process View : Tools

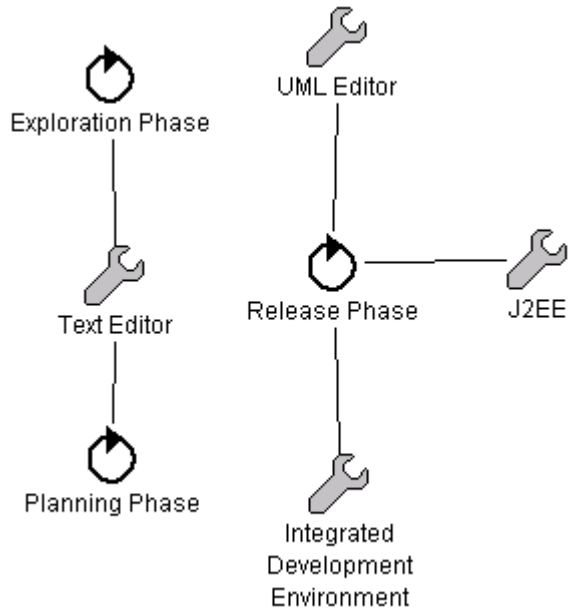

Figure 30. Process view: tools.

4.3 ONLINE-ENTERTAINMENT SERVER /EXPLORATION PHASE

4.3.1 Description

An initial set of requirements is gathered. The development team familiarize themselves with the tools, technology and practices they are to use during the project.

4.3.2 Calendar Time

1. Days taken to complete the activities of this phase. (Begin date - End date)

4.3.3 Effort

1. Time spent in the set of activities for this phase. (Hours)

4.3.4 Input Criteria

Initial set of requests from the customer.

4.3.5 Exit Criteria

1. A completed set requirements described as user stories.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

4.3.6 Product Flow

This activity does not consume any artifacts.
This activity does not modify any artifacts.
This activity produces the following artifacts:

- Requirements Document

4.3.7 Involved Roles

The following roles are involved with this activity:
- Customer
- Project Leader

4.3.8 Used Tools

This activity uses the following tools:
- Text Editor

4.4 *ONLINE-ENTERTAINMENT SERVER / PLANNING PHASE*

4.4.1 Description

Requirements are prioritized, and an overall schedule is made. The content of the first release is agreed upon.

4.4.2 Calendar Time

1. Days taken to complete the activities of this phase. (Begin date - End date)

4.4.3 Effort

1. Time spent in the set of activities for this phase. (Hours)

4.4.4 Input Criteria

1. A completed set of requirements described in the requirements document.

4.4.5 Exit Criteria

1. A completed set of requirements implemented during the iteration.

4.4.6 Product Flow

This activity consumes the following artifacts:
- Release

This activity modifies the following artifacts:
- Requirements Document

This activity does not produce any artifacts.

4.4.7 Involved Roles

The following roles are involved with this activity:
- Developer
- Project Leader

4.4.8 Used Tools

This activity uses the following tools:
- Text Editor

4.5 *ONLINE-ENTERTAINMENT SERVER / RELEASE PHASE*

4.5.1 Description

The release phase is where development is done. The release phase is repeated in two to four week cycles, each iteration bringing new functionality to the system.

4.5.2 Calendar Time

1. Days taken to complete the activities of this phase. (Begin date - End date)

4.5.3 Effort

1. Time spent in the set of activities for this phase. (Hours)

4.5.4 Input Criteria

1. A completed set of requirements implemented during the iteration.

4.5.5 Exit Criteria

1. A completed release that covers the functionality of the selected requirements.

4.5.6 Product Flow

This activity consumes the following artifacts:

- Requirements Document

This activity does not modify any artifacts.
This activity produces the following artifacts:

- Release

4.5.7 Activities

The following are the activities of this phase:

- Analysis
- Design
- Programming
- Continuous integration
- Test system

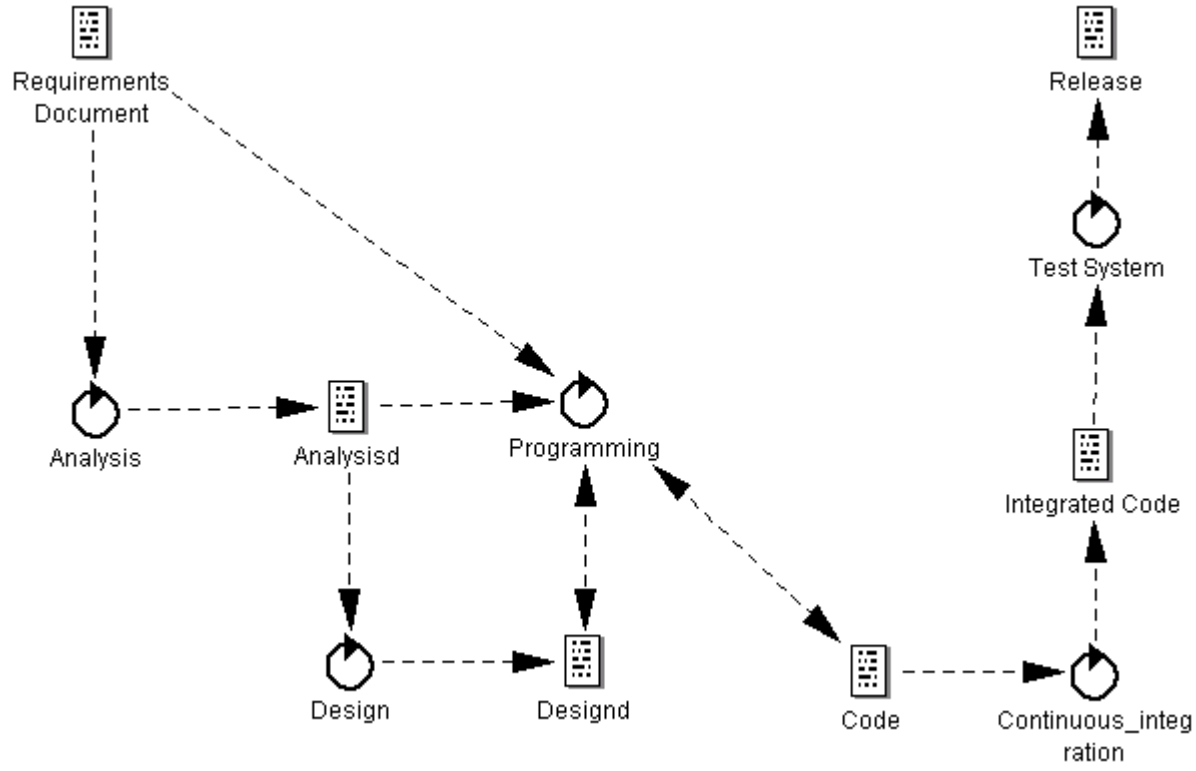

Figure 31. Product flow release phase.

4.5.8 Process View: Roles

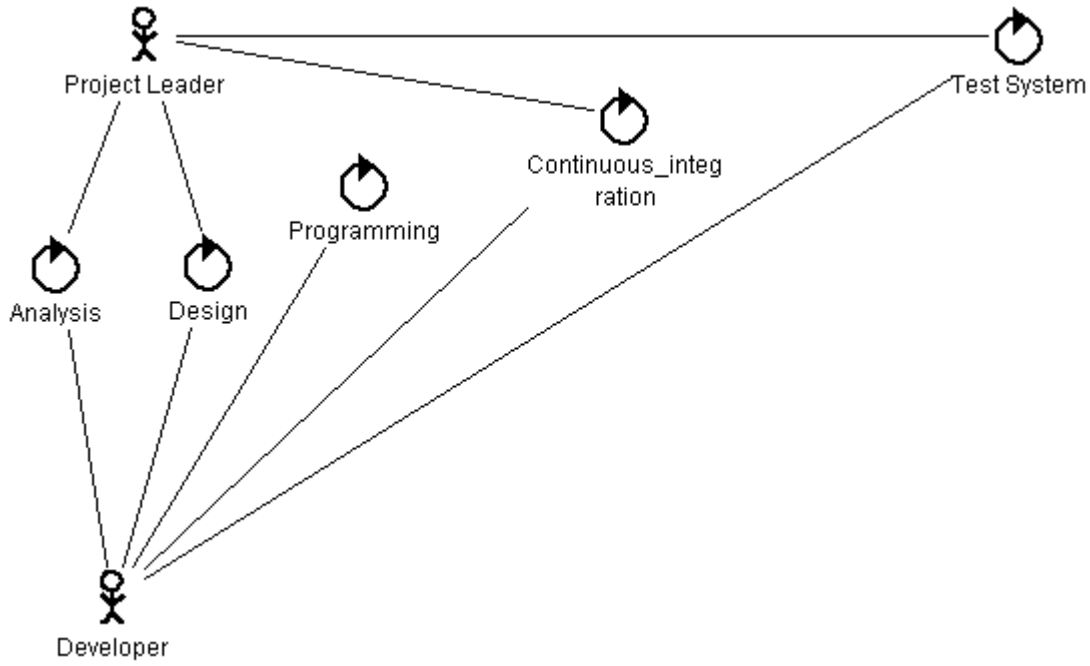

Figure 32. Process view: release phase roles.

4.5.9 Process View: Tools

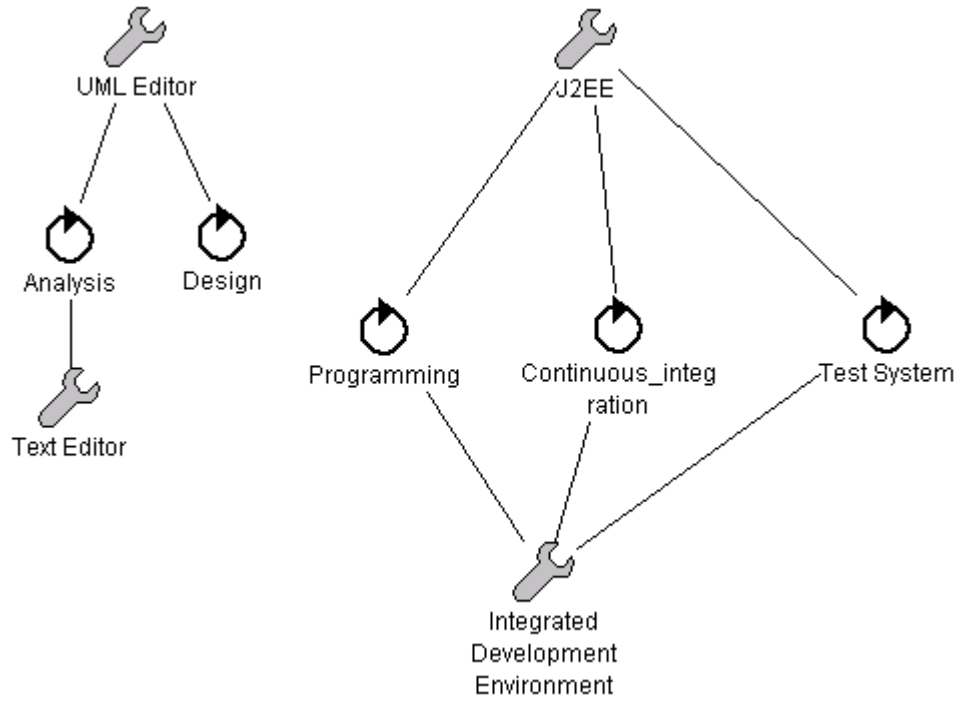

Figure 33. Process view: release phase tools

4.5.10 Involved Roles

The following roles are involved with this activity:
- Developer
- Project Leader

4.5.11 Used Tools

This activity uses the following tools:
- J2EE
- UML Editor
- Integrated Development Environment

4.5.12 Activity: *Analysis*

4.5.12.1 Description

The developer pair analyses the feature to be implemented, and agrees on an overall solution to the problems involved.

4.5.12.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

4.5.12.3 Effort

1. Time spent analyzing the set of selected requirements. (Hours)

4.5.12.4 Input Criteria

1. A completed release that covers the functionality of the selected requirements.

4.5.12.5 Exit Criteria

1. A completed analysis that covers the functionality of the selected requirements.

4.5.12.6 Product Flow

This activity consumes the following artifacts:
- Requirements Document

This activity does not modify any artifacts.
This activity produces the following artifacts:
- Analysisd

4.5.12.7 Involved Roles

The following roles are involved with this activity:
- Developer
- project Leader

4.5.12.8 Used Tools

This activity uses the following tools:
- UML Editor
- Text Editor

4.5.13 Activity: *Design*

4.5.13.1 Description

A coarse sketch is made of the main functionality of the implementation of the feature to be implemented.

4.5.13.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

4.5.13.3 Effort

1. Time spent designing the set of selected requirements.

4.5.13.4 Input Criteria

1. A completed analysis that covers the functionality of the selected requirements.

4.5.13.5 Exit Criteria

1. A completed design that covers the functionality of the selected requirements.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets).

4.5.13.6 Product Flow

This activity consumes the following artifacts:

- Analysisd

This activity does not modify any artifacts.
This activity produces the following artifacts:

- Designd

4.5.13.7 Involved Roles

The following roles are involved with this activity:

- Developer
- Project Leader

4.5.13.8 Used Tools

This activity uses the following tools:

- UML Editor

4.5.14 Activity: *Programming*

4.5.14.1 Description

1. Produce the source code.

4.5.14.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

4.5.14.3 Effort

Time spent coding (modules, functions, methods) (Hours).

1. Time spent in this activity. (Hours)

4.5.14.4 Input Criteria

1. A completed design that covers the functionality of the selected requirements.

4.5.14.5 Exit Criteria

1. Source code (modules, functions, methods).
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

4.5.14.6 Product Flow

This activity consumes the following artifacts:

- Requirements Document
- Analysisd

This activity modifies the following artifacts:

- Designd
- Test Code
- Test Cases
- Code

This activity does not produce any artifacts.

4.5.14.7 Activities

The following are the subactivities of this activity:

- Test First
- Pair Programming
- Test

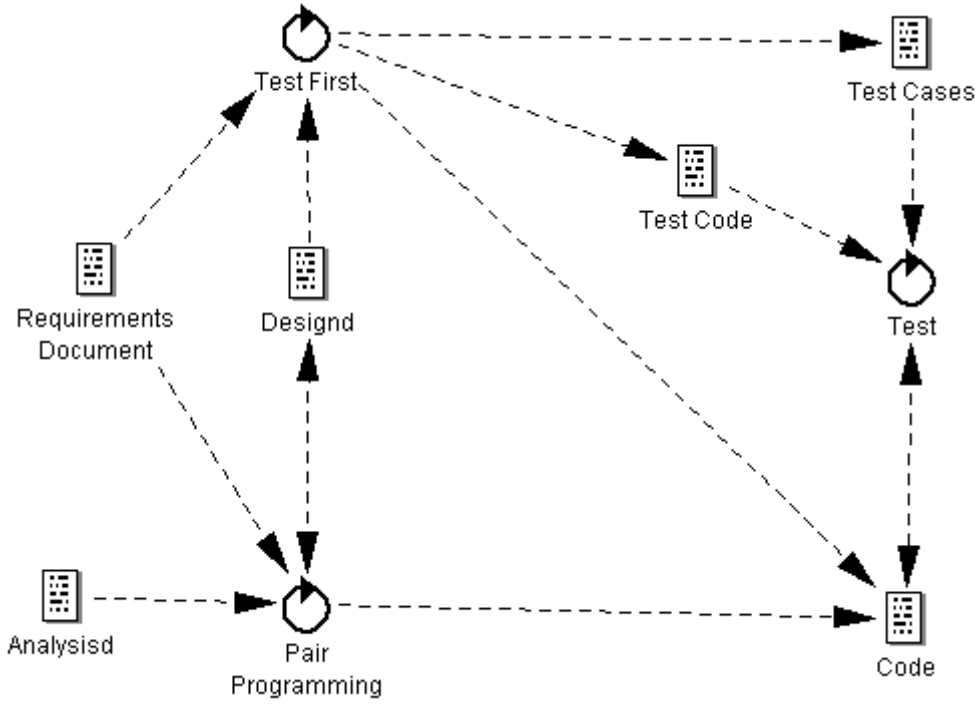

Figure 34. Product flow programming.

4.5.14.8 Process View: Roles

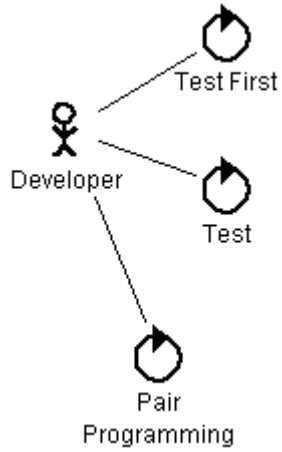

Figure 35. Process view: programming roles.

4.5.14.9 Process View: Tools

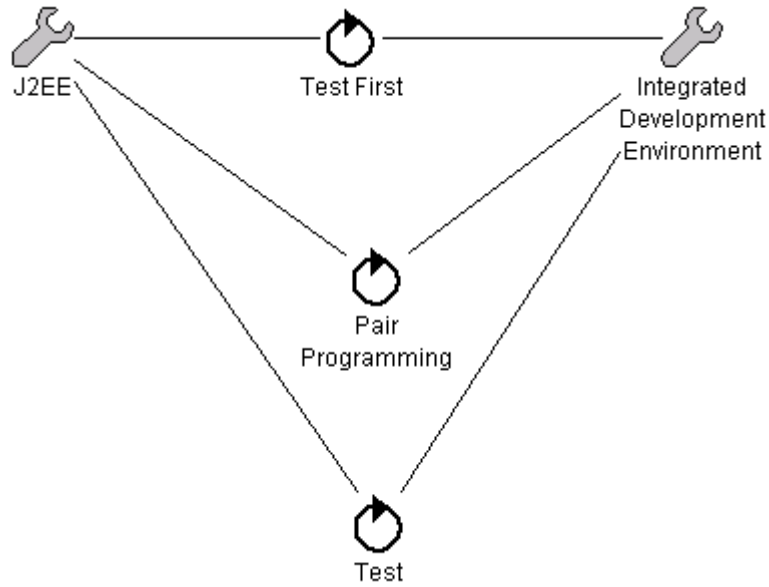

Figure 36. Process view: programming tools.

4.5.14.10 Involved Roles

The following roles are involved with this activity:

- Developer

4.5.14.11 Used Tools

This activity uses the following tools:

- J2EE

- Integrated Development Environment

4.5.14.12 Activity: Test First

4.5.14.12.1 Description

All coding is done so that a unit test is written for the functionality before the actual code. Test-first development is also a design tool.

4.5.14.12.2 Input Criteria

1. A completed design that covers the functionality of the selected requirements.
2. A list of user stories to be implemented during the iteration.

4.5.14.12.3 Exit Criteria

1. Completed collection of classes and methods that will be used to test the code.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets).

4.5.14.12.4 Product Flow

This activity consumes the following artifacts:
- Requirements Document
- Designd

This activity does not modify any artifacts.
This activity produces the following artifacts:
- test Code
- test Cases
- Code

4.5.14.12.5 Involved Roles

The following roles are involved with this activity:
- Developer

4.5.14.12.6 Used Tools

This activity uses the following tools:
- J2EE
- Integrated Development Environment

4.5.14.13 Activity: Pair Programming

4.5.14.13.1 Description

All programming is done in pairs of developers, sitting at one computer.

4.5.14.13.2 Input Criteria

1. A completed design that covers the functionality of the selected requirements.
2. A completed analysis that covers the functionality of the selected requirements.
3. A list of user stories to be implemented during the iteration.

4.5.14.13.3 Exit Criteria

1. Source code (modules, functions, methods).
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

4.5.14.13.4 Product Flow

This activity consumes the following artifacts:
- Requirements Document
- Analysisd
- test Cases

This activity modifies the following artifacts:
- Designd

This activity produces the following artifacts:
- Code

4.5.14.13.5 Involved Roles

The following roles are involved with this activity:
- Developer

4.5.14.13.6 Used Tools

This activity uses the following tools:
- J2EE
- Integrated Development Environment

4.5.14.14 Activity: Test

4.5.14.14.1 Description

Customer-defined acceptance tests and developer-defined unit tests are collected in a test suite, and run as often as necessary.

4.5.14.14.2 Input Criteria

1. Completed collection of classes and methods that will be used to test the code.

4.5.14.14.3 Exit Criteria

1. Tested code (modules, functions, methods).

4.5.14.14.4 Product Flow

This activity consumes the following artifacts:
- Test Code
- Test Cases

This activity modifies the following artifacts:
- Code

This activity does not produce any artifacts.

4.5.14.14.5 Involved Roles

The following roles are involved with this activity:
- Developer

4.5.14.14.6 Used Tools

This activity uses the following tools:
- J2EE
- Integrated Development Environment

4.5.15 Activity: *Continuous Integration*

4.5.15.1 Description

The common codebase must always have a running version of the system. Development is done so that any implemented new functionality is integrated into the codebase as soon as it is ready, preferably at least once a day.

4.5.15.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

4.5.15.3 Effort

1. Time spent integrating the code. (Hours)

4.5.15.4 Input Criteria

1. Tested code (modules, functions, methods).

4.5.15.5 Exit Criteria

1. An integrated and tested product
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

4.5.15.6 Product Flow

This activity consumes the following artifacts:
- Code

This activity does not modify any artifacts.
This activity produces the following artifacts:
- Integrated Code

4.5.15.7 Involved Roles

The following roles are involved with this activity:
- Developer
- Project Leader

4.5.15.8 Used Tools

This activity uses the following tools:
- J2EE
- Integrated Development Environment

4.5.16 Activity: *Tests System*

4.5.16.1 Description

All customer-defined acceptance tests are run at the end of the iteration. The customer then accepts the release, or points out areas that need to be fixed.

4.5.16.2 Calendar Time

1. Days taken to complete this activity. (Begin date - End date)

4.5.16.3 Effort

1. Time spent in the set of activities for this phase. (Hours)

4.5.16.4 Input Criteria

1. An integrated and tested product.

4.5.16.5 Exit Criteria

1. A released and integrated product, which covers the functionality of the entire system.
2. Calendar time, effort, and produced artifacts size data. (Data collection sheets)

4.5.16.6 Product Flow

This activity consumes the following artifacts:
- Integrated Code

This activity does not modify any artifacts.
This activity produces the following artifacts:
- Release

4.5.16.7 Involved Roles

The following roles are involved with this activity:
- developer
- Project Leader

4.5.16.8 Used Tools

This activity uses the following tools:
- J2EE
- Integrated Development Environment

4.6 ONLINE-ENTERTAINMENT CLIENT/ARTIFACTS

4.6.1 Artifact: *Requirements Document*

4.6.1.1 Description

System Requirements Specification
Sample of the contents:
0. Table of contents
1. Introduction
2. Service architecture: It provides the general framework on which the service is based, and general constraints to be considered.
3. Service functional requirements: describe requirements at the level of the entire service.
4. Client side functional requirements: The use cases of the client side of the service are listed. For each use case, associated functional requirements are listed.
5. Server side functional requirements: The use cases of the server side of the service are listed. For each use case, associated functional requirements are listed.
6. Performance and usability requirements. For example: Mobile device memory limitation. Examples of usability and performance requirements are:
- Requirement 001: Midlets at runtime shall not require more than 662Kb. For mobile devices, runtime memory availability may be even lower.
- Requirement 002: A dedicated thread should be responsible for network access and management of the user, so the user can continue interacting with the device at all times.

4.6.1.2 Product Flow

This artifact is produced by the following activities:
- Exploration Phase

This artifact is modified by the following activities.
- Planning Phase

This artifact is used by the following activities:
- Release Phase
- Analysis
- Programming
- Test First
- Pair Programming

4.6.2 Artifact: *Analysisd*

4.6.2.1 Description

The conceptual design of the application, i.e., the problem divided into logical blocks.

4.6.2.2 Product Flow

This artifact is produced by the following activities:
- Analysis

This artifact is not modified by any activity.
This artifact is used by the following activities:
- Design
- Programming
- Pair Programming

4.6.3 *Artifact*: *Designd*

4.6.3.1 Description

This artifact contains the design of:
The set of classes that are part of each component, and its attributes, the interaction among classes in order to provided the desired functionality

4.6.3.2 Product Flow

This artifact is produced by the following activities:
- Design

This artifact is modified by the following activities.
- Programming
- Pair Programming

This artifact is used by the following activities:
- Test First

4.6.4 Artifact: *Test_Code*

4.6.4.1 Description

Collection of classes and methods that will be used to test the code not yet built or in progress. This will allow the developer to do "Test first by intention", and drive all the task development testing-first.

4.6.4.2 Product Flow

This artifact is produced by the following activities:
- Test First

This artifact is modified by the following activities.
- Programming

This artifact is used by the following activities:
- Test

4.6.5 Artifact: *Test Cases*

4.6.5.1 Description

A TestCase is defined as a group of tests that have the same fixtures. Each test is a method on a TestCase class. Every time a test is run, all fixtures are set up before (created) and torn down (destroyed) afterwards. This is to make sure that no single test will influence another. If you want one situation to influence another, you should put it into a single test. Note that a fixture could also be a specific database setup that could be torn down by rolling back or doing a transaction in reverse.
-- http://community.borland.com/article/0,1410,20645,00.html

4.6.5.2 Product Flow

This artifact is produced by the following activities:
- Test First

This artifact is modified by the following activities.
- Programming

This artifact is used by the following activities:
- Pair Programming
- Test

4.6.6 Artifact: *Code*

4.6.6.1 Description

Lines of code written that conforms to the coding standard

4.6.6.2 Product Flow

This artifact is produced by the following activities:
- Test First
- Pair Programming

This artifact is modified by the following activities.
- Programming
- Test

This artifact is used by the following activities:
- Continuous Integration

4.6.7 Artifact: *Integrated Code*

4.6.7.1 Description

Integrated code that conforms to the coding standard.

4.6.7.2 Product Flow

This artifact is produced by the following activities:
- Continuous Integration

This artifact is not modified by any activity.
This artifact is used by the following activities:
- Test System

4.6.8 *Artifact*: *Release*

4.6.8.1 Description

Integrated and tested code that conforms to the coding standard.
According to the customer negotiation, it includes all the features and stories agreed for the current release as well as documentation, user manuals, etc.

4.6.8.2 Product Flow

This artifact is produced by the following activities:
- Release Phase
- Test System

This artifact is not modified by any activity.
This artifact is used by the following activities:
- Planning Phase

4.7 *ONLINE-ENTERTAINMENT CLIENT/ROLES*

4.7.1 *Role*: *Customer*

4.7.1.1 Description

The customer chooses what will deliver business value, chooses what to do first and what to defer, and defines the tests to show that the system does what it needs to.
--Extreme Programming Installed, Chapter 1.

4.7.1.2 Participation

This role participates in the following activities:
- Exploration Phase

4.7.2 *Role*: *Developer*

4.7.2.1 Description

The developers analyze, design, test, program and integrate the system. The developers estimate the difficulty of all stories, and track the peace at which they can deliver stories to the customer.
--Extreme Programming Installed, Chapter 1.
This role is assumed by the members of the project with knowledge and experience in programming Wireless Internet Applications.
The required skills for this domain are:
- Programming skills / Project support skills
- Knowledge of the wireless environment
- Network programming and configuration, or embedded software experience are preferrable

4.7.2.2 Participation

This role participates in the following activities:
- Planning Phase
- Release Phase
- Analysis
- Design
- Programming
- Test First
- Pair Programming
- Test
- Continuous Integration
- Test System

4.7.3 Role: *Project Leader*

4.7.3.1 Description

The project leader brings the customer and developers together and helps them meld into a smoothly operating team. The project leader doesn't do the process; he/she makes the process smoother.
--Extreme Programming Installed, Chapter 1.
Schedules meetings (e.g. Iteration Plan, Commitment Schedule), makes sure the meeting process is followed, and records results of meeting for future reporting. Goes to meetings, brings back useful information, pays for pizza, keeps the rain off, fills out personnel actions.
Does not tell people what to do (Customer and Iteration Plan do that), when to be done (Commitment Schedule), or check to see how they're doing (Tracker).
-- The Wiki Web

4.7.3.2 Participation

This role participates in the following activities:
- Exploration Phase
- Planning Phase
- Release Phase
- Analysis
- Design
- Continuous Integration
- Test System

4.8 *ONLINE-ENTERTAINMENT CLIENT/TOOLS*

4.8.1 Tool: *J2EE*

4.8.1.1 Description

J2EE technology and its component based model simplify enterprise development and deployment. The J2EE platform manages the infrastructure and supports the Web services to enable development of secure, robust and interoperable business applications. The J2EE platform is the foundation technology of the Sun ONE platform and Sun's Web services strategy
Taken from http://java.sun.com/j2ee/

4.8.1.2 Usage

This tool is used by the following activities:
- Release Phase
- Programming

- Test First
- Pair Programming
- Test
- Continuous Integration
- Test System

4.8.2 Tool: *UML Editor*

4.8.2.1 Description

Tool for creating, and maintaining documents whose contents are standard UML-Diagrams.

4.8.2.2 Usage

This tool is used by the following activities:

- Release Phase
- Analysis
- Design

4.8.3 Tool: *Integrated Development Environment*

4.8.3.1 Description

An integrated development environment (IDE) is a programming environment that has been packaged as an application program, typically consisting of a code editor, a compiler, a debugger, and a graphical user interface (GUI) builder. The IDE may be a standalone application or may be included as part of one or more existing and compatible applications.
Taken from http://search390.techtarget.com/gDefinition/0,294236,sid10_gci754848,00.html
Important integrated environment for Java are:
Visual Age for Java
JBuilder
JDK
Sun Studio: This environment was improved for using J2ME.

4.8.3.2 Usage

This tool is used by the following activities:

- Release Phase
- Programming
- Test First
- Pair Programming
- Test
- Continuous Integration
- Test System

4.8.4 Tool: *Text Editor*

4.8.4.1 Usage

This tool is used by the following activities:

- Exploration Phase
- Planning Phase
- Analysis