Deployment Package
Highly Iterative Software Process for the Entry Profile

Notes:
This document is the intellectual propriety of its author’s organization. However, information contained in this document is free of use. The distribution of all or parts of this document is authorized for non-commercial use as long as the following legal notice is mentioned:

© Lappeenranta University of Technology

Commercial use of this document is strictly forbidden. This document is distributed in order to enhance exchange of technical and scientific information.

This material is furnished on an “as-is” basis. The author(s) make(s) no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material.

The process described in this Deployment Package is not intended to preclude or discourage the use of additional processes that Very Small Entities may find useful.

Author(s)
J. Kasurinen, Lappeenranta University of Technology (LUT), (Finland)

Editor(s)
Claude Y. LAPORTE – École de technologie supérieure (ÉTS), (Canada)

Creation date 24/04/2014

Last update 01/03/2015

Version 1.0
## Version History

<table>
<thead>
<tr>
<th>Date (dd/mm/yyyy)</th>
<th>Version</th>
<th>Description</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>24/04/2014</td>
<td>0.1</td>
<td>Document creation</td>
<td>J. Kasurinen</td>
</tr>
<tr>
<td>14/05/2014</td>
<td>0.2</td>
<td>First full draft of the document</td>
<td>J. Kasurinen</td>
</tr>
<tr>
<td>09/06/2014</td>
<td>0.3</td>
<td>Second draft based on editorial comments.</td>
<td>J.Kasurinen, Claude Y. Laporte</td>
</tr>
<tr>
<td>06/08/2014</td>
<td>0.4</td>
<td>Third draft based on editorial comments.</td>
<td>J. Kasurinen, Claude Y. Laporte</td>
</tr>
<tr>
<td>25/11/2014</td>
<td>0.5</td>
<td>Release candidate 1</td>
<td>J. Kasurinen, Claude Y. Laporte</td>
</tr>
<tr>
<td>01/03/2015</td>
<td>1.0</td>
<td>1st published version</td>
<td>J. Kasurinen</td>
</tr>
</tbody>
</table>

## Abbreviations/Acronyms

<table>
<thead>
<tr>
<th>Abre./Acro.</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLC</td>
<td>Downloadable content – additional content for products, usually games, which is sold separately from the main product.</td>
</tr>
<tr>
<td>DP</td>
<td>Deployment Package - a set of artefacts developed to facilitate the implementation of a set of practices, of the selected framework, in a Very Small Entity.</td>
</tr>
<tr>
<td>HISP</td>
<td>Highly Iterative Software Process – the model described in this deployment package</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization (<a href="http://www.iso.org">http://www.iso.org</a>)</td>
</tr>
<tr>
<td>RUP</td>
<td><em>Rational Unified Process</em></td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>SY</td>
<td>System (composed of hardware elements and oftenly of software elements</td>
</tr>
<tr>
<td>TR</td>
<td>Technical Report</td>
</tr>
<tr>
<td>VSE</td>
<td>Very Small Entity – enterprise, organization, department or project having up to 25 people.</td>
</tr>
<tr>
<td>VSEs</td>
<td>Very Small Entities</td>
</tr>
</tbody>
</table>

© J. Kasurinen
Table of Contents

1. Technical Description ........................................................................................................ 4
   Purpose of this document .................................................................................................. 4
   Why this topic is Important? ........................................................................................... 4

2. Definitions .......................................................................................................................... 6
   Generic Terms .................................................................................................................... 6
   Specific Terms .................................................................................................................... 6

3. Relationships with ISO/IEC 29110 ................................................................................. 9

4. Description of Activities, Tasks, Steps, Roles and Products ........................................... 11
   4.1 Project Management Process ....................................................................................... 11
      4.1.2 Activity PM.0 Proof of Concept Planning .......................................................... 12
      4.1.3 PM.1 Project Planning Activity ........................................................................... 15
      4.1.4 PM.2 Project Plan Execution Activity ............................................................... 15
      4.1.5 PM.3 Project Assessment and Control Activity ................................................ 16
      4.1.6 PM.4 Project Closure Activity .......................................................................... 18
   4.2 Software Implementation Process .............................................................................. 21
      4.2.1 SI.0 Initial Software Requirements Analysis and Design Activity .................... 22
      4.2.2 SI.1 Software Implementation Initiation Activity .............................................. 24
      4.2.3 SI.2 Software Components Identification Activity .......................................... 24
      4.2.4 SI.3 Software Construction Activity ................................................................ 26
      4.2.6 SI.4 Software Integration and Tests Activity ................................................... 26
      4.2.7 SI.5 Software Requirements Analysis Activity .............................................. 28
      4.2.8 SI.6 Product Acceptance Tests Activity ............................................................ 30
      4.2.9 SI.7 Product Delivery and Release Activity ..................................................... 30
   4.3 Role Description ............................................................................................................ 31
   4.4 Product Description ...................................................................................................... 32

5. Templates ........................................................................................................................... 35
   5.1 Developing Mobile Game with HISP: Iterations and their main objectives .......... 35
   5.2 Example of Main Design Document Table of Contents for a Game Product ........ 37
   5.3 Example of the Proof of Concept document Table of Contents ............................. 38

6. Example of Activity Lifecycle .......................................................................................... 39

7. Checklist ............................................................................................................................ 42
   7.1 Main Design Document Checklist for Requirements .............................................. 42
   7.2 Product Delivery Checklist for Acceptance Testing ................................................ 43

8. Tools ................................................................................................................................. 44

9. References ........................................................................................................................ 46

10. Evaluation Form .............................................................................................................. 47

© J. Kasurinen
1. Technical Description

Purpose of this document
This Deployment Package (DP) supports the Generic profile group of ISO/IEC 29110 for software development [ISO/IEC 29110]. The Generic profile group is composed of 4 profiles: Entry, Basic, Intermediate and Advanced. The Generic profile group has been developed for Very Small Entities (VSEs) involved in the development of non-critical system (i.e. composed of hardware elements and oftenly of software elements) or software.

A DP is a set of artefacts developed to facilitate the implementation of a set of practices in a Very Small Entity (VSE). A DP is not a process reference model (i.e. it is not prescriptive). The elements of a typical DP are: description of processes, activities, tasks, roles and products, template, checklist, example, reference and reference to standards and models, and tools.

The content of this document is not normative, it is entirely informative.

This document is intended to be used by a VSE to establish processes to implement any development approach or methodology including, e.g., agile, evolutionary, incremental, test driven development, etc. based on the organization or project needs of a VSE.


The Generic profile group does not imply any specific application domain, although the empirical research and process modelling work in which this deployment package is based on is aimed towards mobile application developers and games development industry. This DP applies to non-critical software development projects with high amounts of iteration and need to adjust the product design based on the customer feedback or creative aspects, which cannot be identified, verified or validated prior to the development of a functional prototype.

Why this topic is Important?
The software process, from an initial concept to a verified and validated release version, is the fundamental process that software-producing VSE has to have. However, in some domains the plan-driven process models such as waterfall or RUP-based approaches can hardly be used, because the product design or required features cannot be identified or designed comprehensively beforehand (e.g. when a project starts). In these domains the development process often requires a functional prototype, proof-of-concept design or other form of user input to validate the requirements, and it is also expected that the product design and final features will change and mature during the development process.

This DP is designed to assist these types of VSEs in implementing the following features to principles of the Management and engineering guide of the Entry Profile [ISO/IEC 29110:2012]:

- Design which is not finalized before the implementation.
- Starting the development work with varying amounts of details in the initial design.

© J. Kasurinen
Changes in the product are allowed in any part of the project besides acceptance testing.

Support for highly iterative approaches to the design-develop-test-cycle.

Support the design phase with a separate development process which produces the proof-of-concept prototype.

Divide the main software implementation process to three main phases: predesign (preproduction), development (production) and delivery (post-production).

This DP defines objectives, activities, tasks, roles and products for the project management and software implementation processes. The aim of this DP is not to replace, but to supplement the Entry Profile. This DP describes the additions needed to achieve the desired features.
2. Definitions

In this section, the reader will find two sets of definitions. The first set defines the terms used in all Deployment Packages, i.e. generic terms. The second set of terms used in this Deployment package, i.e. specific terms.

**Generic Terms**

*Process:* set of interrelated or interacting activities which transform inputs into outputs [ISO/IEC/IEEE 12207].

*Activity:* a set of cohesive tasks of a process [ISO/IEC/IEEE 12207].

*Task:* required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a process [ISO/IEC/IEEE 12207].

*Sub-Task:* When a task is complex, it is divided into sub-tasks.

*Step:* one element (numbered list item) in a procedure that tells a user to perform an action (or actions) [ISO/IEC 26514]. In a deployment package, a task is decomposed in a sequence of steps.

*Role:* a defined function to be performed by a project team member, such as testing, filing, inspecting, coding. [ISO/IEC/IEEE 24765]

*Product:* piece of information or deliverable that can be produced (not mandatory) by one or several tasks. (e. g. design document, source code).

*Artefact:* information, which is not listed in ISO/IEC 29110 Part 5, but can help a VSE during the execution of a project.

*System:* combination of interacting elements organized to achieve one or more stated purposes. [ISO/IEC/IEEE 15288]

*Software:* all or part of the programs, procedures, rules, and associated documentation of an information processing system. [ISO/IEC 2382-1]

**Specific Terms**

*Added content:* additional content or features of the product developed and released after the first release version of the product. See Downloadable content.
Alpha testing: A verification and validation phase where the product system is tested with all functionalities and all intended features and content. Contextually similar to system testing. [Adapted from ISO/IEC/IEEE 24765]

Asset: Asset means any resource such as source code, script, component, 3D object, sound effect file, graphical texture or other file or part which is used in the product to provide functionality or content. [Adapted from ISO/IEC/IEEE 24765]

Beta testing: A verification and validation phase where system testing is conducted on fully operational product to assess if the product is ready for delivery. Can also be characterized as an internal acceptance testing by the developer organization. [Adapted from ISO/IEC/IEEE 24765]

Component: set of functional services in the software, which, when implemented, represents a well-defined set of functions and is distinguishable by a unique name [ISO/IEC 29881:2008]

Delivery: release of the product to the third party or other organization, which manages the final acceptance testing, distribution, sales and marketing of the developed product. [Adapted from ISO/IEC/IEEE 24765]

Development process: The entire chain of activities, phases and tasks performed by the organization from the initial statement of work and design to the released version of the product. [Adapted from ISO/IEC/IEEE 24765; development]

Downloadable content: separately sold added content or features of the product developed and released after the first release version of the product. Especially in the games industry known by the acronym DLC.

Feature Freeze: a declared state on development work, during which no new features or content is allowed into the product. Technical development work, such as debugging or code optimization is usually allowed to enhance the product stability. [Adapted from ISO/IEC/IEEE 24765]

Interest group: A group of people or organization, which has personal or financial interest on the product-in-development. For example target customers, intended audience, potential publishers or potential financers. [Adapted from ISO/IEC/IEEE 24765; stakeholder]

Main design document: A document which outlines the main features and intended usage of the product and if applicable, the genre, storyboard, visual theme, main characters and other central content of the intended product. In games industry, it is informally called 'Game Design Document'. [Adapted from ISO/IEC/IEEE 24765; design description]

Preproduction: a development process phase including set of activities, which precedes the commitment to the development of the entire product, with the intent of producing proof-of-concept prototype or comprehensive design to study the feasibility of the proposed software product. [Callele et al.: 2005, Kanode and Haddad: 2009]
**Product:** the result of the development process, can be either packaged off-the-shelf software, or software released as a service. [Adapted from ISO/IEC/IEEE 24765]

**Production:** a development process phase including set of activities, which include all of the process activities needed to develop the product from the initial design to the deliverable product. [Callele et al.: 2005, Kanode and Haddad: 2009]

**Proof-of-Concept (prototype):** A quickly built version of product, which demonstrates the main features, user interface, visual concepts or functionalities of the proposed product. Does not actually have to be a functional version, can also be only a mock-up with fake back-end services. [Adapted from ISO/IEC/IEEE 24765; prototype]

**Post-production:** a development process phase including set of activities in which the deliverable product is finalized, validated and verified against the release requirements by third party and packaged for commercial release. [For example Kanode and Haddad: 2009]

**Release version:** the version of the product made publicly available or put to commercial use. [Adapted from ISO/IEC/IEEE 24765; release]
3. Relationships with ISO/IEC 29110


This DP introduces a general model for organizing the process activities in management and software implementation in a way which is conceptually compatible with the principles of the Entry Profile. This deployment package extends, and only minimally modifies the activities and tasks of the Entry Profile. Table 1 lists, in italic in the left column, the activities, of the Entry profile modified/added in this DP.

<table>
<thead>
<tr>
<th>Activities of this Deployment Package</th>
<th>Activities of the Entry Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM.0 Proof of Concept Planning</td>
<td></td>
</tr>
<tr>
<td>PM.1 Project Planning</td>
<td>PM.1 Project Planning</td>
</tr>
<tr>
<td>PM.2 Project Plan Execution</td>
<td>PM.2 Project Plan Execution</td>
</tr>
<tr>
<td>PM.3 Project Assessment and Control</td>
<td>PM.3 Project Assessment and Control</td>
</tr>
<tr>
<td>PM.4 Project Closure</td>
<td>PM.4 Project Closure</td>
</tr>
<tr>
<td>SI.0 Initial Software Requirements</td>
<td>SI.2 Software Requirements</td>
</tr>
<tr>
<td>Analysis and Design</td>
<td>Analysis</td>
</tr>
<tr>
<td>SI.1 Software Implementation Initiation</td>
<td>SI.1 Software Implementation</td>
</tr>
<tr>
<td>SI.2 Software Component Identification</td>
<td>SI.3 Software Component</td>
</tr>
<tr>
<td>SI.3 Software Construction</td>
<td>SI.4 Software Construction</td>
</tr>
<tr>
<td>SI.4 Software Integration and Tests</td>
<td>SI.5 Software Integration and</td>
</tr>
<tr>
<td>SI.5 Software Requirements Analysis</td>
<td>Tests</td>
</tr>
<tr>
<td>SI.6 Product Acceptance Tests</td>
<td>SI.2 Software Requirements</td>
</tr>
<tr>
<td>SI.7 Product Delivery and Release</td>
<td>SI.6 Product Delivery</td>
</tr>
</tbody>
</table>

Table 1. Mapping between the activities of the Entry profile extended in the DP

The objectives and tasks defined in section 4 of this DP extend the objectives and tasks of the corresponding activity of the Entry model. In a few occasions, the objectives and tasks (e.g. SI.3 and SI.6) are so close to the Entry Profile that no additional tasks are defined.

Figure 1 illustrates the 3 phases of the Highly Iterative Software Process with the added activities.
Activities happen between two processes: Project Management and Software Implementation. The development process has three phases: Proof of Concept Development (preproduction), Production (Implementation) and Post-production and Delivery. During Implementation, the process is highly iterative, iterating between plan-do-check-act-activities in design-develop-test-assess cycles.
4. Description of Activities, Tasks, Steps, Roles and Products

4.1 Project Management Process
Figure 2 presents the 3 development process phases, the flow of information and the main activities in the Project Management process in this DP.

![Figure 2 Project Management process diagram with main activities, flow of information and process phases](image)

© J. Kasurinen
The purpose of the Project Management (PM) process is to establish and carry out in a systematic way the tasks of the software implementation process, which allows complying with the project’s objectives, expected quality, time and costs. The PM process is divided to three phases, which illustrate the project management tasks before, during and after the development process of the product.

The 3 phases of the PM Process are:

- Proof of Concept Development (preproduction) phase:
  - PM.0 Proof of Concept Planning

- Production phase:
  - PM.1 Project Planning
  - PM.2 Project Plan Execution
  - PM.3 Project Assessment and Control

- Post-production and Delivery phase
  - PM.4 Project Closure and Delivery

The project management process is similar to PM process of the Entry profile. This DP documents only the differences and additions to the Entry Profile. The first project management activity (i.e. Proof of Concepts Planning) has been added to the Entry Profile.

Tip: For more complete view into the entire process, getting familiar with the Entry Profile is highly recommended.

4.1.2 Activity PM.0 Proof of Concept Planning

This activity provides:

- Develop the Initial Concept internally by using any design method available (for example team discussions, pitching contest or brainstorming).

- Review the Initial Concept for sanity and validate it with Proof of Concept.

- Accept or Reject the Proof of Concept and if accepted, document the initial version of Main Design Document.

- Make the decision to commit to the development process and advance to the Production phase or restart the design work for Initial Concept.
### Proof of Concept Planning Process

#### Objectives:

The primary objective of the Proof of Concept Planning Process is to produce and communicate an effective and workable project plan with the first design for the developed product explained in the first version of the Main Design Document.

This process determines the scope of the project management and technical activities, identifies process outputs, project tasks and deliverables, establishes schedules for project tasks, required resources to accomplish project tasks and provides the development team with the first actual product design for both technical aspects and the intended features and content.

#### Rationale:

Whatever the size of the project, good planning is essential if it is to succeed. Effective software project management depends on thoroughly planning the progress and milestones of a project. A plan formulated at the start of a project should act as a driver for the project and basis for the for the validation tests later with the test users. The initial plan should be detailed enough so that the first functional version can be developed based on it, but also allow later revisions and changes according to the collected feedback.

#### Roles:

- Project Manager
- Work Team
<table>
<thead>
<tr>
<th>Step Description</th>
<th>Artifacts</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1. Conduct the Initial Design</strong></td>
<td>Product Designer</td>
<td>Step 1. Conduct the Initial Design</td>
</tr>
<tr>
<td></td>
<td>Organizational Management Team</td>
<td>Step 2. Develop the Proof-of-Concept prototype</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Step 3. Decide on the commitment to work</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Step 4. Create more detailed schedule and project plan</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Artifacts:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project Plan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Product Design Document</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proof-of-Concept Prototype</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proof-of-Concept Plan</td>
</tr>
<tr>
<td><strong>Step 2. Develop the Proof-of-Concept prototype</strong></td>
<td><strong>Steps:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Step 1. Conduct the Initial Design</td>
<td>Step 1. Conduct the Initial Design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The product designer identifies the desired features and product content and collects the first set of requirements with the project manager and work team. Design includes features, target platform, hardware solutions and – if applicable – genre, visual direction and theme.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Step 2. Develop the Proof-of-Concept prototype</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The work team develops a proof-of-concept prototype of the designed product, which would typically include things like</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• demonstration of the most important features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• demonstration of the visual direction and theme</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• demonstration of the usability with the selected devices</td>
</tr>
<tr>
<td></td>
<td></td>
<td>At this point the product may still lack all own assets, being solely based on the placeholder objects and quickly developed code derived from the organization’s earlier projects.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Step 3. Decide on the commitment to work</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Based on the proof-of-concept prototype, the organizational management team makes the decision if the product is developed to a full product, based both on technical and economical points of view. If decided to commit to work, OMT formally assigns – “greenlights” - the work team to develop the product further.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Tip: the organizational management team should not be hesitant to reject designs at this point. In many organizations - especially in the game industry - several different setups and variations may be tested before the one selected for becoming a full product is found.</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Step 4. Create more detailed schedule and project plan</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>After the commitment is made, the actual project plan including tasks, work packages, technical modules and artistic assets is made and mapped against the time and resources. The initial concept is clarified and finalized to be the first version of the main design document and the project plan is written and distributed to the work team.</td>
</tr>
</tbody>
</table>
4.1.3 PM.1 Project Planning Activity
This activity is the same as in the Entry profile. Since a Proof of Concept has been developed and approved at the previous activity, this document is used as an input to task PM1.1.

4.1.4 PM.2 Project Plan Execution Activity
In addition of Project Plan Execution activities and tasks defined in the Entry profile, this activity provides following:
— Monitoring the project against the Main Design document.
— Review and Revise the Main Design document against collected feedback and test data.

<table>
<thead>
<tr>
<th>Role</th>
<th>Task List</th>
<th>Input Products</th>
<th>Output Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>PJM, WT</td>
<td>PM.2.3 Monitor and record status of the Project Plan execution against the Main Design document.</td>
<td>Main Design Document</td>
<td>Progress Status Record</td>
</tr>
<tr>
<td>PJM, CUS, WT, OMT</td>
<td>PM.2.4 Review and revise the main design document against the collected feedback from the representatives of the target customer segment or other interest group.</td>
<td>Main Design Document Change request</td>
<td>Main Design Document (revised)</td>
</tr>
</tbody>
</table>

Project Plan Execution

<table>
<thead>
<tr>
<th>Objectives:</th>
<th>To implement and update the actual work tasks of the project in accordance with the project plan and design documentation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rationale:</td>
<td>Ideally when the project plan has been agreed and communicated to all team members, work of the development of the product which is the subject of the project should commence. Since the product needs to be verified and validated against the customer needs and interests, during the implementation also reviews and revisions of the project plan and main design document should be conducted.</td>
</tr>
</tbody>
</table>
| Roles:              | Project Manager  
|                     | Work Team  
|                     | Customer  
|                     | Organizational Management Team |
| Artifacts:          | Main Design document  
|                     | Progress status reports |
| Steps:              | Step 1. Develop a new version of the product based on the design  
|                     | Step 2. Submit the newest version for testing  
|                     | Step 3. Review and revise the design |
### Step Description:

#### Step 1. Develop a new version of the product based on the design

Work team develops a new version of the product based on the main design document. The new version does not have to be a complete, fully functional product. The additions and changes between the different versions may be something like additional content, new feature, changes to the earlier development, or system optimization to enhance performance.

#### Step 2. Submit the newest version for testing

After the intended amount of development is completed, a feature freeze should be declared so that the current solutions can be tested and feedback can be collected from the test users and other interest groups.

*Tip:* The interest groups can be any organizations or partners who are willing to give feedback: test users, random target customer representatives, work crew conducting explorative testing, funding organizations, customers, publisher or others.

#### Step 3. Review and revise the design

When change needs or deviations between the documentation and actual product, or needs based on collected feedback, are identified, the main design document is revised. These revisions are included to the product from the next version cycle onwards.

*Tip:* On some occasions the revision needs may involve major changes to the product, which should be approved by the organizational management team. Additionally, added content design such as DLC feature sets are also formed from these review and revise –cycles.

### 4.1.5 PM.3 Project Assessment and Control Activity

In addition of Project assessment and control activities and tasks defined in the Entry profile, this activity provides:

- Evaluation of the current version of the product against the interest group concerns.
- Decision if the product is mature enough for delivery
### Role

<table>
<thead>
<tr>
<th>Role</th>
<th>Task List</th>
<th>Input Products</th>
<th>Output Products</th>
</tr>
</thead>
</table>
| PJM WT IGR | PM.3.4 Evaluate the newest version of the product with the interest groups:  
- current content  
- current features  
- usability  
- learnability  
- change needs based on collected feedback  
- appropriateness  
- user experience, “is the product fun to use” | Main design document  
Software Configuration [most recent from project repository] | Change Request |

**Tip:** Evaluation here can be conducted for example with usability tests with the interest group members, user testing, explorative testing or market interviews. The important aspect of this task is to collect the evidence on if the product fulfils the customer needs, or what should be changed to better fulfil the customer needs.

| PJM WT OM | PM.3.5 Evaluate if the product is ready for delivery.  
**Tip:** This should be done in parallel with the task SI.6.1 to establish that the product does not have any major change needs, and that the current version fulfils the quality criteria. | Software Configuration [most recent from project repository]  
Test Report [most recent] | |

### Project Assessment and Control Process

**Objectives:**
The purpose of the Project Assessment and Control is to determine the status of the project, ensure that the project performs according to plans and schedules, within projected budgets and it satisfies technical objectives and that the final product matches the desires of the target audience and other interest groups.

This process includes redirecting the project activities, as appropriate, to correct identified deviations and variations from other project management or technical processes and conducting validation and verification actions periodically to ensure that the product matches the intended usage.

**Rationale:**
A project plan is a document that can be used to guide the execution of a project while main design document is used to define the features and content. Unless the actual performance of the execution of the project is tracked against the plan, the plan will have limited value beyond the initiation of the project. If the features and content are not periodically tested with interest group representatives, the product may not be
commercially as successful as it should have been.

**Roles:**
- Project Manager
- Work Team
- Interest Group Representatives
- Organizational Management Team

**Artifacts:**
- Main design document
- Software Configuration (newest prototype of the product from the project repository)
- Change Requests
- Test Plan

**Steps:**
- Step 1. Collect interest group data
- Step 2. Identify plan deviations and change needs
- Step 3. Process change requests

**Step Description:**

**Step 1. Collect interest group data**
Periodically the product prototype should be reviewed by the interest group representatives against the current design and to collect further design and content change requests.

**Step 2. Identify plan deviations and change needs**
Based on any deviations and needs discovered during the collect interest group data activity, it may be necessary to identify and evaluate significant costs, schedules and technical needs to implement changes.

**Step 3. Process change requests**
Any change request must be managed and controlled, as there will be an impact on the project plan, schedule, amount of redundant work and costs. Typically for a change request the following steps should be undertaken:
- Estimate the technical and artistic effort to implement change
- Re-estimate the project schedule and costs
- Estimate how the changes affect existing assets

If no major needs are identified or all new change requests are unfeasible to implement, the organizational management team may also decide to shift the product from development to delivery and post-production phase.

---

**4.1.6 PM.4 Project Closure Activity**
The Project Closure activity provides the project’s documentation and products in accordance with the product requirements, including final activities needed for the developed product to be finalized, delivered and published. In addition of Project closure activities defined in the Entry profile, this activity provides:

- Support of product acceptance testing, required verification and validation

© J. Kasurinen
— Definition of a plan for the product support lifecycle

<table>
<thead>
<tr>
<th>Role</th>
<th>Task List</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>PJM</td>
<td>PM.4.3 Deliver product for acceptance testing.</td>
<td>Project Plan</td>
<td>Acceptance Record</td>
</tr>
<tr>
<td>IGR</td>
<td></td>
<td>Main Design Document</td>
<td></td>
</tr>
<tr>
<td>WG</td>
<td></td>
<td>Software Configuration [ready to be delivered]</td>
<td></td>
</tr>
<tr>
<td>OM</td>
<td></td>
<td>Change Request</td>
<td></td>
</tr>
<tr>
<td>PJM</td>
<td>PM.4.4 Conduct changes needed for product acceptance.</td>
<td>Software Configuration [rejected]</td>
<td>Software Configuration [ready to be delivered]</td>
</tr>
<tr>
<td>WG</td>
<td></td>
<td>Change Request</td>
<td></td>
</tr>
<tr>
<td>PM</td>
<td>PM.4.5. Define support and post-production development life-cycle model for the product.</td>
<td>Project Plan</td>
<td>Acceptance record</td>
</tr>
<tr>
<td>OM</td>
<td></td>
<td>Main design document</td>
<td>[updated]</td>
</tr>
<tr>
<td>IGR</td>
<td></td>
<td></td>
<td>Statement of Work [follow-up project]</td>
</tr>
</tbody>
</table>

**Project Closure**

<table>
<thead>
<tr>
<th>Task Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Objectives:</strong></td>
<td>Project Closure typically involves releasing the final deliverables to the customer or publisher, handing over project documentation to the business, conducting final configurations and revisions to fulfil the acceptance testing needs and finally, making the decisions on how to maintain the product after the initial release.</td>
</tr>
<tr>
<td><strong>Rationale:</strong></td>
<td>A project closure process ensures that all project outputs are delivered correctly and that the follow-up projects have access to all of the project-related material.</td>
</tr>
<tr>
<td><strong>Roles:</strong></td>
<td>Project Manager</td>
</tr>
<tr>
<td></td>
<td>Work Team</td>
</tr>
<tr>
<td></td>
<td>Organizational Management Team</td>
</tr>
<tr>
<td></td>
<td>Interest Group Representatives</td>
</tr>
<tr>
<td><strong>Artifacts:</strong></td>
<td>Project plan</td>
</tr>
<tr>
<td></td>
<td>Software configuration (Product)</td>
</tr>
<tr>
<td></td>
<td>Acceptance record</td>
</tr>
<tr>
<td></td>
<td>Main design document</td>
</tr>
<tr>
<td><strong>Steps:</strong></td>
<td>Step 1. Deliver product to acceptance testing</td>
</tr>
<tr>
<td></td>
<td>Step 2. Pass verification and validation</td>
</tr>
<tr>
<td>Step Description:</td>
<td>Step 3. Define how the product support and post-production life-cycle will be managed.</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 1. Deliver product to acceptance testing</strong></td>
<td>The software system and associated documentation is delivered to the organization responsible for acceptance testing for commercial release.</td>
</tr>
<tr>
<td><strong>Step 2. Pass verification and validation</strong></td>
<td>Complete the final revisions and change requests imposed by the acceptance testing organization to pass verification and validation.</td>
</tr>
</tbody>
</table>
| **Step 3. Define how the product support and post-production life-cycle will be managed.** | As there may be multiple versions of the product over time and/or continued product maintenance, it is necessary to formally record all intended continuation plans and added content follow-up projects before committing to other projects.  
*Tip:* Although the added content such as DLC is usually designed during the main development and is a part of the main design document, this step makes the formal decision on what parts of those designed additions will be produced in the follow-up projects. |
4.2 Software Implementation Process

Figure 3 presents the 3 development process phases, the flow of information and the activities of the Software Implementation (SI) process of this DP.

The purpose of the Software Implementation (SI) process is the systematic performance of the analysis, design, construction, integration and tests activities for new or modified systems.
software products according to the specified requirements.

The 3 phases of the SI Process has the following activities:

- **Proof of Concept Development** (pre-production) phase:
  - SI.0 Initial Software Requirements Analysis and Design

- **Production phase** (iterating until requirements satisfied):
  - SI.1 Software Implementation Initiation
  - SI.2 Software Component Identification
  - SI.3 Software Construction
  - SI.4 Software Integration and Tests
  - SI.5 Software Requirements Analysis

- **Post-production and Delivery phase**:
  - SI.6 Product Acceptance Tests
  - SI.7 Product Delivery and Release

The SI process includes several activities which are similar to the Entry profile. This DP documents only the differences and additions. *Tip: For more complete view into the entire process, getting familiar with the Entry Profile is highly recommended.*

Generally the concept in this DP is that a proof of Concept is developed during the first activity of the SI process. Once the Proof of Concept has been approved, the production phase is an iteration cycle, which allows product design to become more detailed during the development work and allow starting the development activities with a less-than-complete requirements specification and an initial version of a project plan. If the development work of the production phase does not have any iterations, the SI process is conducted in a same way as in the entry profile.

### 4.2.1 SI.0 Initial Software Requirements Analysis and Design Activity

The Initial Software Requirements Analysis and Design activity analyses the initial customer requirements and product design, and establishes the first version of the project software requirements. In addition to the tasks described in the Software Requirements Analysis activity of the Entry profile, this activity provides:

- Development of the *Proof-of-Concept* prototype for the product
- Elicitation, analysis and specification of the first version of the *Main Design Document*
<table>
<thead>
<tr>
<th>Role</th>
<th>Task List</th>
<th>Input Products</th>
<th>Output Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>PJM</td>
<td>SI.0.1 Develop the <em>Proof-of-concept prototype</em> from the <em>Initial Concept</em>. The development work is mostly using placeholder components and simplified assets to test out and demonstrate that the proposed concept can work with the technology in the selected system platform.</td>
<td><em>Initial Concept</em></td>
<td><em>Proof-of-Concept prototype</em></td>
</tr>
<tr>
<td>PD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGR</td>
<td>SI.0.2 Document or update the <em>Main Design Document</em>. Identify and consult information sources such as customers, marketing or designers to compile the <em>Main Design Document</em> which identifies main features, functionality and assets.</td>
<td><em>Proof-of-Concept prototype</em></td>
<td><em>Main Design Document [initial]</em></td>
</tr>
<tr>
<td>PD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PJM</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Initial Software Requirements Analysis and Design**

**Objectives:**

The objective of this activity is to clearly define the scope of the project, provide the first tangible design and identify the key requirements of the system.

**Rationale:**

It is important to clearly define the project scope (boundaries) and to identify key functionalities of the future system to avoid problems like forgotten key functionalities, requirements creep or misunderstood customer preferences.

**Roles:**

- Work Team
- Interest Group Representatives
- Organizational Management Team
- Project Manager
- Product Designer

**Artifacts:**

- Initial concept
- Proof-of-Concept prototype
- Main design document

**Steps:**

- Step 1. Produce the *Initial Concept* with appropriate approach
- Step 2. Produce the *Proof-of-Concept prototype* of the product
- Step 3. Create the first version of the *Main Design Document*

**Step Description:**

**Step 1. Produce the Initial Concept with appropriate approach**

During this Step, the work team captures the key concepts of the business domain of the customer and creates a new design which is considered economically feasible. The interest group representatives may be used to assist the work team and designers by collecting...
information (existing documentation or explanation) that will facilitate this understanding.

**Step 2. Produce the Proof-of-Concept prototype of the product**

From the initial design, Work Team creates the first proof-of-concept prototype of the product for the organizational management team to review and assess. At this point the development work is done with simplified models and placeholder content.

**Step 3. Create the first version of the Main Design Document**

Having produced a satisfactory proof-of-concept the development work is approved for a full product, and the work team with product designer should define the first version of the Main Design Document from the Initial Concept and Proof-of-Concept prototype. The Main Design Documents holds up to date information on the product main features, technical decisions, operating platforms, developed assets and if appropriate, genre, story outline, concept art and other creative design work for the product. See part 5. Templates for an example.

### 4.2.2 SI.1 Software Implementation Initiation Activity

The SI Initiation activity ensures that the *Project Plan* established in Project Planning activity and the *Main design document* created in the *Initial Software Requirements Analysis and Design* is committed to by the Work Team. The activity provides:

- Review of the *Project Plan* and *Main design document* by the Work Team to determine task assignment.
- An implementation environment established.
- Existing documents and assets of the product are retrieved from the project repository if the project is a follow-up project on existing product.

The tasks and steps in this activity are the same than the tasks and steps of the Entry model.

### 4.2.3 SI.2 Software Components Identification Activity

The Software Component Identification activity transforms the software requirements and design plans into the architecture of system components and creative assets. In addition to the tasks listed in the Entry profile, this activity provides:

- Identify the needed creative assets for this iteration, and then schedule the development of these assets.
- Identify the needed outsourced creative assets, and then schedule the acquisition activities (shopping, bidding, invite tenders etc.) for these assets.
### Software components identification

**Objectives:**
To identify the software components and needed assets that will answer the requirements defined by the design documents, that will be tested against customer expectations before being released and to verify that every requirement is fulfilled.

**Rationale:**
Software components identification is a keystone of a software project. Failure to describe a design architecture that will incorporate all the requirements is a recipe for disaster. In this activity the needs of the development work should be identified so that all designed components and assets are created, but that the amount of redundant work is kept minimal.

**Roles:**
- Work Team
- Project Manager
- Organizational Management Team

**Products:**
- Requirement specifications
- Software Configuration

**Artifacts:**
- Main design document
- Project Plan

**Steps:**
1. Understand project plan and main design document
2. Assign and document the required assets

**Step Description:**

#### Step 1. Understand the project plan and main design document
- Examine each requirement and objective for this development iteration and be sure they are understood and addressed in the design documents.
- If needed, update the Requirements Specifications to add necessary clarification.
  - Store updated document in repository

#### Step 2. Assign and document the required assets
- Analyse the Requirements Specification to generate the needed assets during this iteration.
If needed, describe in detail the appearance and the behaviour of the assets, based on the Requirements Specification in a way that resources for its implementation can be foreseen.

- Provide the details of assets based on the Main Design Document to allow the construction in an evident way.

### 4.2.4 SI.3 Software Construction Activity

The activity Software construction develops the software code, creative assets and data from the Software Components identified in SI.2.

This activity happens as described in the Entry model with the added observation that some work team members may produce only creative assets (graphics, models, sounds etc.). For these assets the testing activity is not conducted, but they are reviewed and approved for acceptable quality by the project manager.

### 4.2.6 SI.4 Software Integration and Tests Activity

The Software Integration and Test activity ensures that the integrated software components satisfy the software requirements, set quality criteria and provide the intended user experience. In addition to the tasks defined in the Entry profile, this activity provides:

- Work Team review of the Main Design Document to determine preferred user experience.

- Collecting feedback from the different interest group representative

- Validating product requirements against the collected data.

<table>
<thead>
<tr>
<th>Role</th>
<th>Task List</th>
<th>Input Products</th>
<th>Output Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>PJM</td>
<td>SI.4.7 Assign feedback collection tasks to the work team members according to the current Project Plan.</td>
<td>Project Plan</td>
<td></td>
</tr>
<tr>
<td>WT</td>
<td>SI.4.8 Conduct feedback collection activities as defined in the Main Design Document and Project Plan and document the results.</td>
<td>Project Plan, Main Design Document, Software Configuration (from Project Repository)</td>
<td>Test Report [updated]</td>
</tr>
<tr>
<td>IGR</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Software Integration and Tests

<table>
<thead>
<tr>
<th><strong>Objectives:</strong></th>
<th>First priority is to ensure that the integrated software components satisfy the software requirements. The second priority is to collect target customer needs and to collect possible change requests or additional requirements to enhance the product during following iterations.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rationale:</strong></td>
<td>This allows executing different types of tests and identifying issues that must be corrected by the software development team before the product can be delivered to the acceptance testing.</td>
</tr>
</tbody>
</table>
| **Roles:**       | Project manager  
|                 | Interest Group Representatives  
|                 | Work Team |
| **Products:**    | Test Report |
| **Artifacts:**   | Project plan  
|                 | Main design document  
|                 | Software configuration (newest prototype) |
| **Steps:**       | Step 1: Prepare the environment and participants  
|                 | Step 2: Execute feedback collection  
|                 | Step 3: Document feedback in Test Report |
| **Step Description:** | **Step 1: Prepare the environment and participants**  
| | During this step, the test environment, the tested product configuration and the participants for the feedback collection are prepared. In practice, the work team may for example prepare a data recording system such as feedback form or survey for the participants, develop a separate test build for the testing work and contact relevant interest groups to arrange schedules. |
|                 | **Step 2: Execute feedback collection**  
| | During the test session, all forms of feedback are collected and recorded. This may include components such as  
| | - Interviews, surveys  
| | - Collecting test system usage data  
| | - Test use metrics  
| | - Test user biometrics  
| | - Qualitative data such as comments, requests or open feedback. |
|                 | **Tip:** The feedback may also come from other sources than purposely arranged test sessions. This includes sources such as formal reviews financers, feedback from the publisher, feedback from test customers, feedback from open beta testing etc. |

© J. Kasurinen
### Step 3: Document feedback in Test Report

The work team members responsible for managing feedback must prepare a test report for each testing session. The report must contain a summary of all information collected from this test iteration, and the raw data should be available in the Project Repository for further analysis.

### 4.2.7 SI.5 Software Requirements Analysis Activity

This additional Software Requirements Analysis activity analyses the collected feedback and reviews the Test Report to establish the possible changes to the product requirements, Project Plan or Main Design Document and establishes the validated project software requirements. This activity provides:

- Work Team review of the Test Report to determine change needs in Requirements Specification or Main Design Document.
- Elicitation, analysis and specification of the new and changed requirements
- Verification and validation of fulfilled requirements.

<table>
<thead>
<tr>
<th>Role</th>
<th>Task List</th>
<th>Input Products</th>
<th>Output Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>PJM, WT</td>
<td>SI.5.1 Review the collected data from Test Report and decide on the change needs to the existing product requirements or designs.</td>
<td>Test Report</td>
<td>Project Plan [reviewed]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Main Design Document [reviewed]</td>
</tr>
<tr>
<td>WT, PD, PJM</td>
<td>SI.5.2 Document or update the Requirements Specification, Project Plan and Main Design Document. Identify, define and internally verify the requirements, which need to be changed or added.</td>
<td>Project Plan [reviewed]</td>
<td>Project Plan [revised]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Main Design Document [reviewed]</td>
<td>Main Design Document [reviewed]</td>
</tr>
<tr>
<td>PJM, OM</td>
<td>SI.5.3 Validate and obtain approval of the new versions of Requirements Specification, Project plan and Main Design Document.</td>
<td>Project Plan [revised]</td>
<td>Project Plan [new version]</td>
</tr>
</tbody>
</table>

### Software Requirements Analysis

**Objectives:** The objective of this activity is to update and clearly define the scope and objectives of the project, and identify and update the key requirements of the system.
### Rationale:
It is important to clearly define the project scope and objectives, and if necessary, update and maintain the requirements based on the collected feedback.

### Roles:
- Work Team
- Project Designer
- Project Manager
- Organizational Management Team

### Products:
- Requirements specification

### Artifacts:
- Test report
- Project Plan
- Main Design Document

### Steps:
1. Step 1. Identify and update requirements
2. Step 2. Schedule and prioritize requirements

#### Step Description:

**Step 1. Identify and update requirements**

During this Step, the work team with project manager and product designer identifies
- New requirements
- Changed requirements and
- Completed requirements

from the Test Report, against the Project Plan, Requirements Specification and the Main Design Document. Work Team drafts a new version of the project documentation based on these requirements, and according to the size of the change requests, gets changes approved internally by the project manager or organizational management team.

**Step 2. Schedule and prioritize requirements**

Using requirements identified in the previous Step, the project manager has to organise and structure identified requirements accordingly for next iterations of the development process.

One way to prioritize requirements and change requests could be following:
- ‘High’ – a functionality, feature or requirement that shall be implemented to the product
- ‘Medium’ - a functionality, feature or requirement that should be implemented to the product
- ‘Low’ - a functionality feature or requirement that could be implemented to the product

The output of this Step is a list of requirements that are organized in the Requirements Document and can be used as a basis on the decisions on what to develop during the next development cycle.
4.2.8 SI.6 Product Acceptance Tests Activity

The Product Acceptance Tests is an activity that ensures that the integrated software components satisfy the software requirements from the viewpoint of the customer, publisher or other external validator. The product acceptance tests provides following objective:

- Interest Group Representatives working as external validators conduct the quality assurance and acceptance testing tasks they consider necessary.

- Work Team integrates Software Components, corrects defects and documents results as instructed by the external validator such as publisher, customer or financer.

Basically this activity is the same as the “Software Integration and Test” activity from the Entry profile, with the added emphasis that the tasks SI.5.4 and SI.5.5 are approved by the Customer, or other Interest Group Representative, such as publisher or financer, to ensure that the integrated software components satisfy the software requirements.

4.2.9 SI.7 Product Delivery and Release Activity

The Product Delivery activity provides the software product to the customers or publishing organization for commercial release. Besides the activities mentioned in the Entry profile for the "Product Delivery" activity, this activity provides:

- Work Team review on follow-up projects and continuance plans.

<table>
<thead>
<tr>
<th>Role</th>
<th>Task List</th>
<th>Input Products</th>
<th>Output Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>PJM</td>
<td>SI.6.4 Perform review of the project outcome and reach consensus on follow-up-projects.</td>
<td>Software Configuration, Main Design Document, Project plan</td>
<td>Initial concept [follow-up project]</td>
</tr>
<tr>
<td>WT</td>
<td></td>
<td></td>
<td>Statement of Work [follow-up project]</td>
</tr>
<tr>
<td>OM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGR</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PRODUCT DELIVERY AND RELEASE

Objectives: To conduct managed delivery and release of the product, and reach consensus on how the product support life-cycle and follow-up projects which will be based on the project outcomes.

Rationale: To ensure that the customer or interest group representatives are given all of the relevant information on the launched product, and that the developing organization has access to all resources planned for follow-up-projects and support life-cycle for this product.

Roles: Work Team

- Project Manager
- Organizational Management Team
- Interest Group Representative

Products:

- Initial concept [follow-up project]
- Statement of Work [follow-up project]
### 4.3 Role Description

This is an alphabetical list of the roles, abbreviations and list of competencies as defined in the engineering and management guide of the Entry profile. In this DP, 3 new roles have been added to the 3 roles already defined in the Entry profile (i.e. Customer, Project Manager, Work Team), with the Entry profile roles being supplemented with the activities from this deployment package. The 3 new roles are: Product Designer, Organizational Management and Interest Group Representatives.

<table>
<thead>
<tr>
<th>Role</th>
<th>Abbreviation</th>
<th>Competency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Project Manager</td>
<td>PJM</td>
<td>Project manager is the person conducting daily management of the product development process and is the supervisor of the work team. Project manager has the ability to make independent decisions on minor changes to the requirements or content of the developed product.</td>
</tr>
<tr>
<td>2. Product Designer</td>
<td>PD</td>
<td>Product designer is a specialist role in the work team, a person or small group of people who are responsible for designing the features and content of the developed product.</td>
</tr>
<tr>
<td>3. Work Team</td>
<td>WT</td>
<td>Work team encompasses the people involved in the product development in a VSE. For example software designers, system architects, developers, programmers, testers and artists are part of the product team.</td>
</tr>
</tbody>
</table>
4. Organizational Management

OMT

Organizational management is the operational management above the project-level, able to make independent business decision on organizational unit level. In a VSE, OM may include all of the owners or executive decision makers, whereas in a large organization, OM may refer to the upper management of one division, a local office, an organization unit, a studio or a subsidiary.

Organizational management is the decision making group, with the absolute authority within the organization to accept or reject proposed designs, and start or stop development projects.

5. Customer

CUS

Customer is the organization or group of individuals, who will be paying for the ability to use or resell the product after successful launch.

6. Interest Group Representatives

IGR

Interest group representatives are the external stakeholders of the development process beyond the work team or organizational unit. Depending on the context, the interest group representatives may include publisher, business partners, financers, target customer segment, external test users or representatives of the organizational body responsible for sales and marketing.

Additionally, people in this role may have advisory or administrative role in the development process, and they either provide feedback for the product enhancement, or validate the product against the requirements for external acceptance.

4.4 Product Description

This is an alphabetical list of the input, output and internal process products not described in the Entry Profile and their descriptions, possible states and the source of the product. Rest of the descriptions (for example for Requirements specification or Change request) are available in the Entry profile.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Source</th>
</tr>
</thead>
</table>
| 1. Initial Concept    | Initial concept is the first description of the product and as a document usually consists of less than 5 pages. Initial concept may have the following characteristics:  
  - Identifies the main features of proposed product.  
  - Identifies the main platforms and form of usage of the proposed product.  
  - Identifies genre and visual style of the proposed product. | PM.0: PD, WT |
## 2. Main Design Document

The main design document is the comprehensive design guide of the product. The main design document contains the most up-to-date information on the following characteristics:

- Defines the main features of the completed product
- Defines the target platforms and forms of usage of the completed product.
- Defines the technical requirements of the completed product
- Identifies the genre and visual style of the completed product.
- Identifies the intended target user groups of the completed product.
- Defines the financial model associated with the completed product.

The applicable statuses are: initial, current, revised.

### Proof-of-Concept

Proof-of-concept prototype is a quickly built version of the product, intended to demonstrate the main features, user interface, visual concepts or...
<table>
<thead>
<tr>
<th>prototype</th>
<th>functionalities of the proposed product.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Does not actually have to be a functional version, can also be only a mock-up with faked back-end services and is mostly used to examine the different designs and proposals for the next product.</td>
</tr>
<tr>
<td></td>
<td>The applicable statuses are: initial, accepted and rejected</td>
</tr>
</tbody>
</table>

5. **Test Report**

In addition to the content defined in the Entry profile, Test Report includes all of the collected test data which can be used to assess the usability, likeability and completeness of the product-in-development. Besides technical items described by the Entry Profile, test report should also include following items:

- Summary of the user tests conducted for the product
- Summary of the user test results
- Identify the location of the collected data from user testing
- Other interest group testing information (from surveys, feedback, open comments)

The applicable statuses are: most recent and updated

<table>
<thead>
<tr>
<th>SI.4</th>
<th></th>
</tr>
</thead>
</table>
5. Templates

5.1 Developing Mobile Game with HISP: Iterations and their main objectives

Iterations and their main objectives with highly iterative software process model (each iteration is typically 1 week long)

- **Month 1**
  1. Initial design, rejected 1. prototype
  2. Revised initial design, rejected 2. prototype
  3. Revised 2. design, accepted 3. prototype; proof-of-concept, project plan internally “greenlighted” for development.
  4. 1st version of the technical part of Main Design Document decided and internally accepted.

- **Month 2**
  5. 1st version of the financial part of Main Design Document decided and internally accepted.
  6. 1st version of the creative part of Main Design Document decided and internally accepted.
  7. All technical infrastructure set up and functional.
  8. Main control schemes, basic product framework operational.

- **Month 3**
  9. First usable features with commercial grade content.
  10. First playable content available, testing started.
  11. Content and features development, review and revisions.
  12. Content and features development, review and revisions.

- **Month 4**
  13. Content and features development, review and revisions.
  14. Content and features development, review and revisions.
  16. Content and feature tuning.

- **Month 5**
  17. Content and feature tuning.
  18. Content and feature tuning.
  19. Content and feature tuning, accepted for delivery.
  20. Delivery: Acceptance testing

© J. Kasurinen
• **Month 6**
  21. Acceptance testing
  22. Acceptance testing
  23. Launch and Development process closure
  24. Development process closure, decisions on maintenance, follow-up-projects.
5.2 Example of Main Design Document Table of Contents for a Game Product

1. Executive Summary, Quick overview
2. Target Audience
3. Main Characters
4. Main Features
   4.1 Main mechanics
   4.2 Movement
   4.3 Physics
   4.4 Multiplayer mode
5. Genre, Setting, Concept Art book*
6. Enemies, NPCs, Other objects
7. Story board, script*
   7.1 Story overview
   7.2 Progression, World 1

   ...
   7.9 Progression, World 8
8. Technical definitions, Tech guide*
   8.1 Platforms, versions
   8.2 Control Scheme
   8.3 Limitations
9. Business definitions*
   9.1 In-app purchases
   9.2 DLC packs
10. Outsourced/Bought Assets

Topics with (*) extended to separate detailed documents, in this document only summarized
5.3 Example of the Proof of Concept document Table of Contents

1. Quick overview
2. Main features, target audience, “statement of content”
   What do you do in the game/with the product, what makes it different from the
   competition, why does it work, why does it sell, what activities are included in the
   product.
3. Visual theme and main story elements, main characters.
4. Cost/profit analysis
   Based on comparable products and available sales figures.
5. Target platform and control scheme, other technical requirements

Entire document possible to summarize to less than five pages.
6. Example of Activity Lifecycle

This section provides visual guidelines for the activity lifecycle in the different phases of the product development as described in this Deployment Package: proof-of-concept development in Figure 4, production in Figure 5 and post-production and delivery in Figure 6.

Figure 4: Example of proof-of-concept phase process activity lifecycle
Figure 5: Example of production phase process activity lifecycle in one iteration
Figure 6: Example of post-production and delivery phase process activity lifecycle
7. Checklist

General checklists of the deployment package Software Implementation – Entry Profile should be applied when appropriate during the development to assess feasibility of the requirements, software components, software integration and testing work and product delivery.

### 7.1 Main Design Document Checklist for Requirements

This Requirement checklist is adapted from [Constr07]. The requirements should be reviewed on each iteration to check that they still fulfil the checklist. Functional requirements (usability, security, reliability, supportability...) and constraints (design, implementation, physical...) follow the guidelines set by the ISO/IEC 29110 principles. Non-functional requirements are inspected and reviewed based on their feasibility and collected information during quality assurance and testing work.

<table>
<thead>
<tr>
<th>Checklist</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RS 1 Testable</strong></td>
<td>All functional requirements are verifiable (objectively)</td>
</tr>
<tr>
<td><strong>RS 2 Complete</strong></td>
<td>Are the functional requirements complete?</td>
</tr>
<tr>
<td><strong>RS 3 Traceable</strong></td>
<td>All functional requirements must be traceable to a systems specification, contractual/proposal clause.</td>
</tr>
<tr>
<td><strong>RS 4 Correct</strong></td>
<td>Functional requirements must be correct (i.e. reflect exactly customer’s requirements)</td>
</tr>
<tr>
<td><strong>RS 5 Unique</strong></td>
<td>Functional requirements must be stated only once</td>
</tr>
<tr>
<td><strong>RS 6 Elementary</strong></td>
<td>Functional requirements must be broken into their most elementary form</td>
</tr>
<tr>
<td><strong>RS 8 High Level</strong></td>
<td>Functional requirement must be stated in terms of final need, not perceived means (solutions)</td>
</tr>
<tr>
<td><strong>RS 9 Quality</strong></td>
<td>Quality attributes have been defined.</td>
</tr>
<tr>
<td><strong>RS 10 Unambiguous</strong></td>
<td></td>
</tr>
<tr>
<td><strong>RS 11 Hardware</strong></td>
<td>Hardware environment is completely defined.</td>
</tr>
<tr>
<td><strong>RS 12 Solid</strong></td>
<td>Technical requirements are a solid base for design</td>
</tr>
<tr>
<td><strong>RS 13 Verification</strong></td>
<td>Testing the product against the requirements must verify that the product is built correctly.</td>
</tr>
<tr>
<td><strong>RS 14 Validation</strong></td>
<td>Testing the product against the requirements must validate that the product is designed correctly for its intended usage and its intended Customer. (Project is building the right product.)</td>
</tr>
<tr>
<td><strong>RS 15 User-based</strong></td>
<td>The non-functional requirements capture the needs and interests of the Customer as closely as is reasonable to expect. (for example, <em>is the user-experience appropriate for the theme? Is the game fun to play?</em>)</td>
</tr>
</tbody>
</table>
7.2 Product Delivery Checklist for Acceptance Testing

This checklist has the most common requirements for product delivery from development to acceptance testing and release [modified from ISO/IEC TR 29110:2012].

**Typical Acceptance Criteria for a VSE**

- The product supporting media is labelled correctly, showing at a minimum product name, release date, and correct version number.
- Product labelling, including delivery location and product acceptance personnel (if applicable), is accomplished.
- The software generated from the project repository in accordance with the delivery instructions.
- The software generated from the project repository in accordance with the requirements for entering acceptance testing.
- The software to be delivered is the latest version of the software in the project repository.
- The Version Description Document has been inspected.
- The Version Description Document is included with the supporting media.
- All tests have been performed successfully.
- All placeholder content is removed or replaced with production-quality content.
- All content in the software is reviewed and approved by the Project Manager or Organizational Management.
- All errors have been corrected.
- All documentations have been updated.
- The User's Manual has been inspected.
- The User's Manual is included with the supporting media.
- All topics in the approved Delivery Instructions Forms have been covered.
- The Acceptance Record Form has been updated and ready for signature.
- The information needed for delivery (e.g. Site address, customer representative) have been verified before delivery.
- Your customer has been informed when a delivery will be performed.
- The customer informed you that all his preparations for delivery have been completed.
8. Tools

Example of a Technical Infrastructure in a Very Small Software Project

In the following tables, a simple set of basic development tools and other needed technical infrastructure are listed for a generic game project. In some categories, also generic requirements and common usage objectives are also listed. In table 8, the tools are in rough order of importance for the development team, although this obviously may change between projects and organizations.

<table>
<thead>
<tr>
<th>General activities in project</th>
<th>Preproduction, design phase</th>
<th>Production, implementation phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design and prototyping tools</td>
<td>Documentation tools, Doc. distribution, graphics software, development tools, game engine</td>
<td>Development tools, game engine, graphics software</td>
</tr>
<tr>
<td>Implementation, development tools</td>
<td>Graphics software, development tools, Development platform (game engine), cloud and repository services</td>
<td>Development platform (game engine), development tools, graphics software, audio software</td>
</tr>
<tr>
<td>Project support tools</td>
<td>Documentation, cloud and repository services</td>
<td>Version control, cloud services, file sharing service, ticket management systems</td>
</tr>
</tbody>
</table>

Table 8. The list of development tool need for a generic game software project

The tool categories are listed in Table 9.

<table>
<thead>
<tr>
<th>Category</th>
<th>General requirements</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio software</td>
<td>Output compatible with other development tools.</td>
<td>Design and implementation of all audio assets of the project.</td>
</tr>
<tr>
<td>Cloud- and repository services</td>
<td>Compatible with all applied development platforms.</td>
<td>Storage and retrieval of developed assets.</td>
</tr>
<tr>
<td>Development tools</td>
<td>Able to support all of the programming languages needed in the project.</td>
<td>Programming tasks, source code editing.</td>
</tr>
<tr>
<td>Documentation</td>
<td>The documents themselves.</td>
<td>Storing and delivering project data.</td>
</tr>
<tr>
<td>Documentation distribution tools</td>
<td>Compatible with all applied development platforms. Able to reach all project work team members.</td>
<td>To distribute new, updated and revised documents and ensure that the new versions and revisions in them are acknowledged.</td>
</tr>
<tr>
<td>Documentation tools</td>
<td>Able to produce all of the needed charts, reports and diagrams. Use format which is compatible with all development platforms.</td>
<td>Tools used to write and draw the documents defined and needed during the development project.</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Development platform (Game engine)</td>
<td>Able to emulate the resources and services available in the delivery environment.</td>
<td>Development and testing work to assess usability, functionality, responsiveness and user interface of the product during the development</td>
</tr>
<tr>
<td>Graphics software</td>
<td>Output compatible with other development tools.</td>
<td>Design and implementation of all graphical assets of the project.</td>
</tr>
<tr>
<td>Ticket management system</td>
<td>Compatible with all applied development platforms.</td>
<td>Management of resources, workload of the project personnel and task distribution within work team.</td>
</tr>
<tr>
<td>Version control system</td>
<td>Compatible with all applied development platforms.</td>
<td>Management of different developed, designed and unfinished assets with the ability to manage their development and roll back changes if necessary.</td>
</tr>
</tbody>
</table>

**Table 9 Tool Categories and descriptions**
## 9. References

<table>
<thead>
<tr>
<th>Key</th>
<th>Reference</th>
</tr>
</thead>
</table>
10. Evaluation Form

Deployment Package **Highly Iterative Software Process– Version 0.4**
Your feedback will allow us to improve this Deployment Package, your comments and suggestions are welcomed.

1. **How satisfied are you with the CONTENT of this deployment package?**
   - [ ] Very Satisfied
   - [ ] Satisfied
   - [ ] Neither Satisfied nor Dissatisfied
   - [ ] Dissatisfied
   - [ ] Very Dissatisfied

2. **The sequence in which the topics are discussed, are logical and easy to follow?**
   - [ ] Very Satisfied
   - [ ] Satisfied
   - [ ] Neither Satisfied nor Dissatisfied
   - [ ] Dissatisfied
   - [ ] Very Dissatisfied

3. **How satisfied were you with the APPEARANCE/FORMAT of this deployment package?**
   - [ ] Very Satisfied
   - [ ] Satisfied
   - [ ] Neither Satisfied nor Dissatisfied
   - [ ] Dissatisfied
   - [ ] Very Dissatisfied

4. **Have any unnecessary topics been included? (please describe)**

5. **What missing topic would you like to see in this package? (please describe)**
   - Proposed topic:
   - Rationale for new topic

6. **Any error in this deployment package?**
   - Please indicate:
     - Description of error:
     - Location of error (section #, figure #, table #):

7. **Other feedback or comments:**

8. **Would you recommend this Deployment package to a colleague from another VSE?**
   - [ ] Definitely
   - [ ] Probably
   - [ ] Not Sure
   - [ ] Probably Not
   - [ ] Definitely Not

Optional
- Name: ________________________________
- e-mail address: ________________________________

Email this form to: claudy.laporte@etsmtl.ca or jussi.kasurinen@lut.fi

© J. Kasurinen