Iterative Software Development - from Theory to Practice

Implementing the Unified Software Development Process in RAFAEL

Amir Tomer, Boaz Shani, Ely Bonne
RAFAEL at a Glance

Operations & Logistics
- R. Gal
- Dr. E. Ganani

Business Development & Marketing
- Dr. E. Ganani

Finance (CTO)
- M. Wainer
- Dr. D. Leshem
- L. Srebro

Research & Development

Strategic Programs & Human Resources

Systems Division
- D. Zait
- Dr. E. Yudilevich

Missiles Division
- Dr. G. Barak

Ordnance Systems Division
- M. Kidron

Propulsion & Explosive Systems Division

Commercial Activity

Export Market

2000 Sales (m$)

- Missile Division 400
- Ordnance Systems Division 110
- Propulsion and Explosive Systems Division 55
- Systems Division 140

Europe 27%
Latin America 46%
Asia + Oceania 32%
North America 37%
The Waterfall Model

Royce, 1970

- Requirements
  - Verification
- Specification
  - Verification
- Design
  - Verification
- Implementation
  - Testing
- Integration
  - Testing
- Maintenance
  - Retirement

Development

- Req. Change
  - Verification
What’s wrong with the Waterfall model?

- Document-based verification until late stages
- Attempt to stipulate unstable requirements too early
- Risk mitigation postponed
- Operational problems discovered too late
- Lengthy modification cycles and much rework

The inevitable result...
The Unified Software Development Process

aka RUP

Core Workflows

Process

Supporting

Business Modeling
Requirements
Analysis and Design
Implementation
Test
Deployment
Config. & Change Management
Project Management
Environment

Inception
Elaboration
Construction
Transition

Phases

preliminary iteration
iter. #1
iter. #2
iter. #n
iter. #n+1
iter. #n+2
iter. #m
iter. #m+1

Organization along Time

Organization along Content

USDP in Rafael - 5
Iterative Development Overview

- Transition
  - Operational system builds

- Construction
  - Operational skeletal system

- Elaboration
  - System scope & concept
  - Solution alternatives
  - SW/HW breakdown
  - System architecture

- Inception
  - Solution alternatives
CSCI Level Stages

Operational CSCIs

Source code modules

SAD/SDD
Software Architecture/Design Description

SRS
S/W Req’rmnts Spec.

Elaboration
Construction

8
CSCI Validation

7
Coding & Unit Testing

6
Software Design

5
Software Requirements

Operational CSCIs

Source code modules

SAD/SDD
Software Architecture/Design Description

SRS
S/W Req’rmnts Spec.
Stage 1: System Boundary

- **Purpose**
  - Provide a common understanding about the system
    - client
    - developers
    - other stakeholders

- **Activities**
  - Scope
  - Relationship with external systems
  - Main requirements

- **Products**
  - System charter / Vision document
Stage 2: System Requirements

• **Purpose**
  – Define system requirements
    • at the best level of knowledge

• **Activities**
  – Requirements elicitation from contract, proposal and other documents
  – Requirements classification and prioritization
  – QFD may be utilized

• **Products**
  – SSS = System/Subsystem Specification
    • Use Case modeling recommended
  – Requirements base
Stage 3: System Design

- **Purpose**
  - Provide a stable proposed solution and architecture

- **Activities**
  - System architectural design
  - Hardware/software breakdown
    - including requirements allocation
  - Feasibility tests to select alternatives

- **Products**
  - SSDD = System/Subsystem Design Description
  - ICD = Interface Control Description
    - External / internal interfaces
    - Either separate or included in SSDD
Stage 4: Software Planning

• **Purpose**
  – Generating a general plan for the software development

• **Activities**
  – Risk Analysis
  – Iteration planning
    • Detailed plan for 1st iteration
    • General plan for following iterations
    • Risk allocation to iterations
  – Resource allocation
  – Testing concept and general planning

• **Products**
  – SDP = Software Development Plan
    • Risk table and mitigation plan
    • Appendix: 1st iteration detailed plan
  – STP = Software Test Plan
Stage 5: Software Requirements

• **Purpose**
  - Provide a clear and detailed definition of the software requirements allocated to the appropriate CSCI
    - At the current iteration level

• **Activities**
  - Deriving software requirements from system requirements
  - Building/updating the Use Case model

• **Products**
  - Use Case model
    - Using CASE tools
  - SRS = Software Requirements Specification
    - Functional requirements (Use Cases)
    - Non-functional requirements
  - Software requirements base
    - Preferably as part of the system requirements base
Stage 6: Software Analysis & Design

- **Purpose**
  - Provide an architecture and design, in various aspects, of the current iteration
    - based on previous iterations + current requirements

- **Activities**
  - Analysis and design modeling
    - Using UML models

- **Products**
  - UML models
    - Class model
    - State charts
    - Sequence diagrams
  - SAD = Software Architecture Description
  - STD = Software Test Description
Stage 7: Coding and Unit Testing

• Purpose
  – Implementation of software modules

• Activities
  – Coding
    • Programming, compilation, link
  – Individual module testing
    • Informal documentation
    • Test coverage and completion approved by STL

• Products
  – Approved code modules
Stage 8: CSCI Validation and Approval

• **Purpose**
  – Approve the CSCI’s readiness for system integration

• **Activities**
  – Integrating the modules comprising the CSCI
  – Performing the tests specified in the STD

• **Products**
  – Approved CSCI version, ready for system integration
  – STR = Software Test Report
    • Detailed results of STD tests
Stage 9: System Integration and Testing

• **Purpose**
  – Accomplish the iteration with an operational approved version of the system

• **Activities**
  – Integration
  – System testing, according to system test specifications

• **Products**
  – Approved version of the system, ready for delivery
System Level Reviews

Inception
- 1 System Boundary
- 2 System Requirements
- 3 System Design
- 4 Software Planning

Elaboration
- 9 System Integration

Construction
- 9 System Integration
- CSCIA
- CSCIB
- ... CSCIC
- CSCII
- CSCIN

System Requirements Review
- System Design Review
- Critical Design Review (S/W)
- Test Readiness Review
- Iteration Completion Review

Test Readiness Review
- external review
- peer review

System Integration
- CSCIA
- CSCIB
- ... CSCIC
- CSCII
- CSCIN

Iteration Completion Review

USDP in Rafael - 18
CSCI Level Review

8
CSCI Validation

7
Coding & Unit Testing

selected iterations

6
Software Design

5
Software Requirements

Iteration Plan Review

Software Specification Review

Software Design Review

Elaboration
Construction

external review
peer review
Conclusions

• **USDP-based iterative software development** has many advantages over the waterfall model
  
• The process may be adapted and tailored to host most of MIL-STD-498 terminology
  – Tailoring requires modified templates

• The iterative process complies with RAFAEL software development procedures

• The tailored process is well accepted by engineers, managers and clients
Dr. Amir Tomer  
Director of Systems & Software Engineering Processes  
RAFAEL  
P.O.Box 2250/1P  
Haifa, 31021  
ISRAEL  
(+972)-4-879-5989  
tomera@rafael.co.il