On the efficiency of domain-based COTS product selection method.
Abstract

- The use of commercial-off-the-shelf (COTS) products is becoming a popular software development method.
- Increasing number of available COTS components.
Introduction

- A COTS component is defined as an independent unit that provides a set of related functions and is suitable for reuse.

- CBS
  - COTS Based Systems - Systems that adopt COTS development as much as possible.
  - Faster delivery with lower resource costs.
Selection methods (1)

- Normally classified in two categories:
  - Intuition approach
    - Select components according to their experience and intuition.
  - Direct assessment (DA)
    - Select components directly from their source.
    - Consider all of the descriptions and based on this takes a decision based on their suitability.
    - More objective than intuition approach.
Selection methods (2)

- Indirect methods
  - Developed by the authors of the paper.
  - Makes use of the specific domain model of the intended system.
    - Application specific domains
    - Technical classification
  - DBCS – Domain based COTS selection method.
Selection methods (3)

- COTS selection
  - Best fit strategy
    - Aims at identifying the best COTS product.
  - First fit strategy
    - Aims at identifying the first COTS product that satisfies all of the requirements.
Some DA methods:

- **OTSO** - Off the shelf option.
  - Consists of three phases
    - Searching
    - Screening
    - Evaluation

- **CISD** - COTS based integrated system development.
  - Identification
  - Evaluation
  - Integration
IIDA – Infrastructure incremental development approach.

- This approach combine the classical waterfall and spiral development models.
- Two phases
  - Analysis of prototype
  - Design prototype
BDCS
A domain Model → A CBS
DBCS

- Consists of two phases:
  1. Set-up phase
     - Vendors roll out their COTS, and map them to those modules of the domain that they find applicable.
  2. Selection phase
     (a) The corresponding modules in a domain model are identified for each of the modules of the CBS in question.
(b) Identify the COTS modules that are applicable for mapping the domain model to the COTS modules.

(c) Non-functional properties of the identified COTS modules are assessed.

(d) The most appropriate COTS modules are selected.
Efficiency of the COTS selection methods

- Analyze the efficiency of the domain based COTS product selection method and the DA method.
- Best fit vs. First fit
Development with Off-The-Shelf Components: 10 Facts

Jingyue Li

17 Oct. 2008
Research design

• An industrial survey with 133 completed projects from 127 companies in Norway, Italy, and Germany

• 28 follow-up telephone interviews.

• Focused on process improvement and risk management
Development with OTS components: actors and activities
Factor 1: development process

- Companies use traditional processes enriched with OTS-specific activities to integrate OTS components.
  - Waterfall and evolutionary process are not suitable?
  - 75% chose their main development processes before they thought about using OTS-components
  - Why do not companies adapt the development process?
  - If companies adapt processes, how do they do so?
  - Insights: Familiarity is important
Factor 2: component selection

- Integrators select OTS components informally. They rarely use formal selection procedures.
  - Selected components in an ad-hoc manner, using in-house expertise and/or web-based search engines
  - Why do not companies use formal selection processes?
  - If a formal selection process was applied, what was done?
  - Insights: cost effectiveness
Factor 3: component selection (cont’)

• There is no specific phase of the development process in which integrators select OTS components. Selecting components in early phases has both benefits and challenges.
  – Most integrators selected OTS components in the early phases - prestudy (38%), requirements (30%), overall design (16%)
  – Reasons for and issues pertaining to select OTS in the prestudy phase
  – Reasons for and issues pertaining to select OTS in requirement/design phase
  – Insights: tradeoff between advantages and disadvantages
Factor 4: component integration

- **Component integration**: Estimators use personal experience when they estimate the effort required to integrate components and most of the time they do not estimate accurately. Stakeholder-related factors will affect dramatically the accuracy of estimates.

  - Reasons for inaccurate effort estimation
  - Insights: provider and clients issues must be considered
Factor 5: quality of the integrated system

- **Quality of the integrated system**: Negative effects of OTS components on the quality of the overall system are rare.
  - Reasons for positive feedback on the quality of OTS components
  - Insights: quality assurance effort of the integrator must be counted
Factor 6: OSS and COTS components

• **OSS and COTS components**: Integrators usually used OSS components in the same way as commercial components, i.e. without modification.

  – Reasons for changing the source code
  – Reasons for not changing the source code
  – Insights: commercial vs. non-commercial, long-term application vs. short term application
Factor 7: locating defects

- **Locating defects is difficult:** Although problems with OTS components are rare, the cost of locating (i.e. within or outside OTS components) and debugging defects in OTS-based systems is substantial.
  
  - Reasons for inefficient defect location
  - Insights: make the defect reproducible
Factor 8: relationship with the provider

- **Relationship with the provider:** The relationship with the OTS component provider involves much more than defect fixing during the maintenance phase.
  - Issues related to component providers
    - Selection
    - Integration
    - Maintenance
  - Insights: know the right person
Factor 9: relationship with clients

- **Relationship with the client:** Involving clients in OTS component decisions is rare and sometimes unfeasible

  - Reasons for not involving clients
  - Insights: clarify clients’ interests and technical capabilities
Factor 10: knowledge management

- **Knowledge management**: Knowledge that goes beyond the functional features of OTS components must be managed.
  - Which knowledge needs to be kept and shared?
    - Component itself
    - How to facilitate the integration
    - Stakeholders
  - Which knowledge management mechanisms to choose?
    - Repository, seminars, Wiki, yellow pages
  - Insights: external knowledge share is rare
Conclusions

• Gap between theories and practices
• Issues to be addressed
  – How can providers and integrators share knowledge of OTS components on a global scale?
  – How can people working on the field establish the “who to contact” yellow pages for each OSS project to facilitate support from OSS communities?
Presentation of the article:

Experiences on Product Development with Open Source Software
by Ari Jaks (2007)

Presenter: Ketil Sandanger Velle
Agenda

1. Introduction
2. Software Architecture
3. Community collaboration
4. Benefits of open source
5. Issues and challenges
6. Summary
Introduction

• Nokia 770 & N800 Internet Tablets
  – WLAN with internet use cases like:
    • Voice & video calls
    • Web browsing
    • Messaging
    • Media consumption
  These are all built using Linux and other open source components

• www.maemo.org – web site that supports Internet Tablet development
Software Architecture

- 428 source code packets
- 25% unmodified OSS
- 50% modified OSS
- 25% COTS & Nokia

- EU report (Nokia 770):
  - 15,000,000 lines of code
  - 200,000 by Nokia
  - 1.5% additional investment
Community collaboration [1/2]

• Selecting core components:
  – Technical suitability of all components and subsystems
  – Fit Requirements and hw specifications
  – Good quality
  – “Mature enough” for consumer products
  – License
  – Use GTK+ for graphic environment
Community collaboration [2/2]

- Work tightly with the community:
  Several parts of the code on N800 released before the tablet
Benefits of open source

- Efficiency
- Quality
- Flexibility
- Software licensing
- Future and roadmaps
- Open source and confidentiality
Issues and challenges

• Hacking vs. Stabilizing
• Architecture management
• Community alignment vs. Backwards capability
• Community participation in product integration
• Investing in community work
Summary

- OSS offers time and cost savings in form of:
  - Readily available components and subsystems
  - Available developers
  - Effective development model
- However: The quality of the final product is Nokia’s own responsibility, so some in house development must be done as well