Increase ICT Project Success with Concrete Scope Management

AEMES, Madrid 2007

Carol Dekkers & Pekka Forselius, 4SUM Partners
www.4sumpartners.com
Agenda

1. ICT projects are unique
2. Scope management concepts
3. Northern and Southern SCOPE
4. Scope managers
5. Making change happen
6. Summary
1. Information and Communication Technology (ICT) projects are unique
Definition of an ICT project

A temporary endeavor undertaken to create an unique ICT (software) product or service
# Engineering and ICT projects

<table>
<thead>
<tr>
<th>Engineering</th>
<th>ICT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Industry</strong></td>
<td>Relatively young</td>
</tr>
<tr>
<td><strong>Product</strong></td>
<td>Intangible (software)</td>
</tr>
<tr>
<td><strong>Cost model</strong></td>
<td>? (USD/SLOC?)</td>
</tr>
<tr>
<td><strong>Metrics</strong></td>
<td>Users/developers → differ</td>
</tr>
<tr>
<td>“Project” definition</td>
<td>Hybrid mixture: R&amp;D, development, conversion, enhancement → many s/w</td>
</tr>
<tr>
<td>Stability &amp; costs</td>
<td>Frequent enhancement, maintenance = 8x develop</td>
</tr>
<tr>
<td>Controls, models</td>
<td>Regulatory (e.g., U.S. FDA), CMM®/CMMI®, SPICE</td>
</tr>
</tbody>
</table>

- Engineering: Mature, Tangible (road, building), Standard (USD/km or /m²), Common (m²), Discrete: Construction, renovation, upgrade → 1 building at a time, Some renovations, initial capital cost > maintenance, Building codes, “sealed” engineering blueprints.
- ICT: Relatively young, Intangible (software), ? (USD/SLOC?), Users/developers → differ, Hybrid mixture: R&D, development, conversion, enhancement → many s/w, Frequent enhancement, maintenance = 8x develop, Regulatory (e.g., U.S. FDA), CMM®/CMMI®, SPICE.
ICT project uniqueness

- 1 in 3 projects → Success
- Young industry → Lack of tools, unit costs, lack of historical data and experience
- Intangible product → Difficult to measure
- Hybrid ‘projects’ → PM challenge
- Risks high → materials, intangible product, requirement completeness & stability
Scope related reasons for ICT project failure

- Lack of top management commitment
- Misunderstanding the requirements
- Not managing change properly
- Failure to gain user commitment
- Lack of effective project management skills
- Lack of adequate user involvement
- Failure to manage end user expectations
- Lack of effective project management methodology
- Unclear/misunderstood scope/objectives
- Changing scope/objectives
ICT project breakthroughs

- Australia → Southern SCOPE
- Finland → Northern SCOPE
- Universal applicability
- Based on concrete scope management
- Modeled on PMBOK principles: Initiate, plan, execute, control, close
2. Scope management concepts
Enhanced scope management

1. Initialize & scope
2. Split into manageable sub-projects (as necessary)
3. Quantify & cost (based on currency per unit size)
4. Manage, track, control, and deliver based on 1-3. Change management based on scope and quantification by unit size
FiSMA scope management concept
FiSMA scope management concept

- Initiating project and software
- Estimating cost and duration
- Controlling progress
- Developing software
- Managing changes
- Closing development
3. Northern and Southern SCOPE
Southern SCOPE

The cost overruns for projects using the southernSCOPE method was found to be less than 10% whereas the industry norm was 84%. (ISBSG, 2005)
Northern SCOPE

- Initiating Processes: Initiating project and software
- Estimating Processes: Estimating cost and duration
- Planning Processes: Developing software
- Controlling Processes: Managing changes, Controlling progress
- Executing Processes: Closing development
- Closing Processes: Closing development
## Northern SCOPE
### Types of ICT projects

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CUSTomer specific new development project</td>
</tr>
<tr>
<td>2.</td>
<td>Software PRODuct new development project</td>
</tr>
<tr>
<td>3.</td>
<td>Software VERSIONion enhancement project</td>
</tr>
<tr>
<td>4.</td>
<td>ICT SERVice development project</td>
</tr>
<tr>
<td>5.</td>
<td>PACKage software configuration project</td>
</tr>
<tr>
<td>6.</td>
<td>Data CONVersion project</td>
</tr>
<tr>
<td>7.</td>
<td>Software INTeGration development project</td>
</tr>
</tbody>
</table>
Rules for splitting a project into sub-projects

1. If the program consists of ICT development and other development work, such as manual process development, re-organising staff or technical development, different type of work should be assigned to separate projects.

2. If you apply incremental or iterative development approach, every increment or iteration should be assigned to separate projects.

3. Different type of ICT development work should be assigned to separate projects.

4. If the program must be stopped consciously for long time, for example to wait external decisions, the work before and after the break should be assigned to separate projects.

5. If two parts of product or service development are similar ICT project type but differ from each other by development technology, they should be assigned to separate projects.

6. If two parts of product or service development are similar ICT project type but differ from each other by development environment, they should be assigned to separate projects.

7. If two parts of product or service development are similar ICT project type but differ from each other by development team experience, they should be assigned to separate projects.

8. If two parts of product or service development are similar ICT project type but differ from each other by quality requirements of target result, they should be assigned to separate projects.

9. If two parts of product or service development are similar ICT project type but differ from each other by stakeholder dependencies, they should be assigned to separate projects.

10. If two parts of product or service development are similar ICT project type but differ from each other by risk level, they should be assigned to separate projects.
FiSMA scope management process components

Scope Management Process

Software Requirements Development
- Elicitation of Customer Needs
- Documented Customer/Market Needs
- Analyse and Classify Requirements
- Software Requirements Document

Software Requirements Management
- Software Change Proposal
- Analyse Requirement Change(s)
- Update Software Req’s Document
- Analyse Impacts of Change(s)
- Software Change and Traceability Record

Software Development Project
- Identify analogy situations of sw and projects
- Define Project Scope
- Project Baseline Metrics
- Project Progress and Performance Metrics
- Project Actuals and Results Metrics
- Initiate the Project
- Report Project Execution Progress
- Closure of the project

Links
- Marketing and Sales
- Business Management
- Knowledge Management
- Organisational Learning

Carol Dekkers & Pekka Forselius, 4SUM Partners
www.4sumpartners.com
New!
Non-traditional ICT project tasks

- Division into independent sub-projects
- Functional size measurement
- Baseline project metrics
- Effort, duration, & cost estimates based on historic project actuals
- Feedback loop for change
Introduction to Functional Size Measurement (FSM)

“Function Points are the square feet or square meters for software”

Carol Dekkers
Project requirements - simplified

1. Functional (user) requirements

2. Non-Functional (user) requirements

3. Other requirements

‘User problem(s) to be solved’

“Project” requirements

‘Developer provided technical solution’
1. Functional user requirements

→ **What** the software must do:

- Business processes performed by / supported by the software (user problem to be solved by software)
- Examples: Record air temperature and pressure; Add new customer
- Important for estimating
- Like a floor plan for software: size is “Functional Size”

**Size = functional size (function points FP)**
2. Non-functional user requirements

→ **HOW** software must perform -ilities: (Suitability, Quality, Interoperability, Security, Reliability, Efficiency, Maintainability, Portability…)

- Constrains the software → increases complexity
- Important for estimating
- Like a building code or contract specifications (part of user problem to be solved)

Not part of functional size
3. Other requirements

→ How software will be developed / ‘built’

- Includes: tools, methods, skills, language, WBS tasks, platforms, software, type of project, etc.
- Part of solution
- Important for estimating
- Like “blueprint” drawings

Not part of functional size
Functional size measurement (e.g., Function Points)

- Quantifies “Base functional components (BFC)”
- Number of “points” based on standard methods
- Similar to sizing rooms in a floor plan

International Function Point Users Group (IFPUG) has 5 BFCs:

- Persistent data: ILF, EIF
- Elementary processes: EI, EO, EQ
Functional size measurement

- Five standard ISO/IEC conformant FSM methods:
  - IFPUG 4.1 unadjusted – ISO/IEC 20926
  - COSMIC – ISO/IEC 19761
  - NESMA – ISO/IEC 24570
  - Mark II – ISO/IEC 20968
  - FiSMA 1.1 - ISO/IEC 29881
4. Scope managers
What is a scope manager

The Scope Manager provides **metrics based project governance**. …very successful in objectively quantifying key project attributes … to enable decision making with respect to project estimates and project risk…

**Typically a metrics specialist** who has excellent skills in business analysis, project estimation and functional size measurement.

…**Independent of the project team** … able to report the status of the project objectively without bias, to a management level that has the authority to proceed, change direction or cancel the project.

*(Morris, 2004)*
Customer & supplier views

1. A big problem is that there are very few customers who understand the size of software, even if they are good at specifying requirements. They want fixed price!

2. Due to missing FSM skills, the developers cannot measure or estimate the size of software. They want to be paid based on hours worked.
Scope manager – the Equaliser

• A Scope Manager is neutral software measurement specialist with good understanding of requirements and development work.

• The customer organisation can hire a professional Scope Manager to help from early tendering phase to the end of the development life-cycle.

• **The customer and developer can use unit pricing e.g. €/fp.**
Scope manager’s tasks in theory

1. Help customer to organise the development program into measurable subsystems and projects.
2. Measure the size of each piece of software to be developed.
3. Help customer to answer the questions of candidate suppliers.
4. Baseline the size with the selected supplier.
5. Help customer to control changes and progress.
6. Ensure that the project data will be collected and submitted to a common repository.

<table>
<thead>
<tr>
<th>Phase of Software Acquisition</th>
<th>Tasks of Scope Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEASIBILITY STUDY, initial requirements</td>
<td>Refining requirements, splitting the development program into projects, measuring the functional size, and providing list of functions required.</td>
</tr>
<tr>
<td>INVITE PROPOSALS AND ENGAGE SUPPLIER</td>
<td>Preparing answers to questions presented by the tenderers, if they may be related to functional size.</td>
</tr>
<tr>
<td>REQUIREMENTS SPECIFICATION</td>
<td>Completing the lists of functional user requirements and measuring the baseline functional size of software.</td>
</tr>
<tr>
<td>SOFTWARE DEVELOPMENT</td>
<td>Controlling projects by function and by phase of MORS (Object Breakdown Structure), measuring the delivered functional size monthly.</td>
</tr>
<tr>
<td>DESIGN</td>
<td>Monitoring the impact of planned changes to functional and non-functional user requirements and maintaining the accepted baseline estimates.</td>
</tr>
<tr>
<td>IMPLEMENTATION</td>
<td></td>
</tr>
<tr>
<td>SYSTEM TEST</td>
<td></td>
</tr>
<tr>
<td>INSTALLATION</td>
<td></td>
</tr>
<tr>
<td>PROJECT CLOSURE</td>
<td>Measuring the actual project delivery rates and productivity and collecting the project data to experience databases.</td>
</tr>
</tbody>
</table>

FiSMA Scope Manager rule definition document 2005
Scope manager’s tools in Finland

• Characteristics analysis
  A method and a simple MS Excel tool for analysing and pointing out the characteristics of a large software development program.

• Experience® Pro 3.1
  Packaged software supporting functional size measurement and determination of the quality requirements, as well as providing the related attachment reports for the contracts. The tool includes several project databases to check the reality and feasibility of the target projects.

• MS Word
  Used to provide a 1 page summary of the system size and quality requirements by sub-system.
5. Making change happen
Sustainable change needs **P.O.W.E.R.**

- **P**redisposition → we see success
- **O**utlook → we will do it
- **W**herewithal → we can do it
- **E**valuation → we will measure it
- **R**esources → we commit to it

*Source: C. Dekkers 2000*
“Motivation moves mountains…
Motivation is everything!”

Principle 6 of Tom Gilb's Ten Most Powerful Principles for Quality in (Software and) Software Organizations, 2nd World Congress on Software Quality, Tokyo, September 2000
### Critical success factors of SPI:

<table>
<thead>
<tr>
<th>Pressure for change</th>
<th>Leadership and vision</th>
<th>Capable people</th>
<th>Actionable first steps</th>
<th>Effective rewards</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>Successful implementation</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>Disinterest</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>Evaporation</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>Frustration</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>Disengagement</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>Dissolution</td>
</tr>
</tbody>
</table>

Source: P. Willman
6. Summary
Results speak for themselves
Thank you!

Carol Dekkers
dekkers@qualityplustech.com

Pekka Forselius
pekkaf@sttf.fi