Tids- och kostnadsuppskattning av RUP Projekt

Micael Erneborg
micael.erneborg@kvadrat.se

Patrik Jonsson
patrik.jonsson@it-arkitekterna.se

Content

• Context: RUP & Project Management Models
• Project Phases & Iteration
• Definitions
• What to Estimate
• Estimation Methods overview
• Practical examples
• Next step
Rational Unified Process (RUP)

PROPS is a generic project model, developed by Ericsson Project Management Institute.

PROPS consists of three parts:
- **Blue U**, project management
- **Yellow U**, development process
- **Red tollgates**, commercial decision points
Does not the Iterative Approach simply mean that we do not know what we are doing?

That is the point!

We iterate until we have learn more about:
- Requirements
- Our technical approach
- How skilled we are at developing software according to the process we follow
- Other risks

Probability of a particular TTM

Calendar time

Lack of precisions in estimate of:
- Time to market (TTM)
- Cost

Estimation

Estimate, Guess, Guestimate, …

People are
- Optimistic, Scared, Want to look good,
- Forget the history, Have selective memories, and
- Think in single point numbers

We do better if we
- Get a clear picture
- Use Team and several experts,
- Use range estimates and statistics,
- Use a formal estimation method, and
- Insist on realistic estimates

We assume you’re
- in control of your requirements, and
- have a sound Project Management process

Unit of estimation:
- Sweden: man-hours
- US: size

Tids- och kostnadsuppskattning
av RUP Projekt: # 6
RAIN 2002
What to estimate for?

Everything that moves ☺
- The complete system
  - Rough estimates
- Use Cases
  - A good foundation for estimation. Important to include challenges with the technology
- Use Case Realizations
  - We know more than the requirements
- Subsystems
  - The chosen technology is integrated in the sub-systems.

Real world case 1

- Estimates done with a matrix containing modules (part of the architecture) and use cases
- Estimates in execution
  - Chief architect (1)
  - Module responsibles (2)
  - Estimates 100%-400% larger
- Positive
  - Project Managers can easily plan which use cases to implement
- Negative
  - New method -> hard to determine if the estimates are resonalbe
- No outcome yet
Use Cases Realizations connect Subsystems

Use Case Model

Architecturally Significant Use Cases

Architecturally Significant (parts of) Packages have been Implemented

All Packages Implemented

Estimate also to assess ROI

Effort

Value

Requirement

Release 2

Release 1

Tids- och kostnadsuppskattning
av RUP Projekt: #9
RAIN 2002

© 2001 IT-Arkitekterna & Kvadrat

Tids- och kostnadsuppskattning
av RUP Projekt: #10
RAIN 2002

© 2001 IT-Arkitekterna & Kvadrat
Methods overview

- Expert Judgements
- Analogy
- Delphi
- Algorithmic Models
  - CoCoMo, Function Points, Object Points, UC Points
- Lichtenberg
- Paired Comparison
- Critical Chain

Expert judgements

$123,450
Algorithmic Models

Scaling Factor
Effort Multiplier

\[ PM = A \times \text{Size}^E \times \prod_{i=1}^{n} EM_i + PM_{Auto} \]

- Cost in Effort (Person Months), Calendar Time, or Size (CoCoMo II)
- Function Points, Object Points, Use Case Points, Boehm, …
- Up to within 20% in 70% of the cases

"Lichtenberg-method"

- Estimate with uncertainty:
  - Min, max, most likely
  - Average = \((\text{min}+\text{max}+3\times\text{likely})/5\)
  - Standard deviation = \((\text{max}-\text{min})/5\)
- Successive break-down:
  - Break-down the activities with the largest uncertainty
  - Estimate the parts and add up:
    » Schedules
    » Costs
    » Standard deviations
- The Red/Green world:
  - Ignore risks and crucial uncertainties first, then
  - Add the corresponding activities
Paired Comparison

- Compare pairwise all "objects" that we estimate for X_i and X_j – are they:
  - X_i >> X_j
  - X_i > X_j
  - X_i = X_j
- This yields a relative comparison of sizes
- Estimate the absolute value for a number of "fix points"
- Calculate the total, here:
  \[
  \begin{align*}
  &+350/1,75 \\
  &+3 \times 350 \\
  &+2 \times 475 \\
  &+2 \times 4 \times 475 \\
  &= 6000h
  \end{align*}
  \]

Real world case 2

- Administrative system with 110-120 use cases
- Migrating legacy to modern framework
- Implemented 1 use case and measured time for that
- Arranged all use cases in 3 sets:
  - Complicated
  - Medium
  - Simple
- Compared to implemented use case and developed estimates during Elaboration
- Adjusted new estimates to the fact that next use cases should be quicker since experience with new framework was building
- Presented estimates to management
- Project was cancelled – should have taken to much time and effort!
Critical Chain

- Based on Critical Chain by Goldratt
- Theory of constrains
- Aggressive planning & Reduce Multitasking
- Critical Chain vs. Critical Line
- Buffer management
  - The Project Buffer protects the delivery date
  - The Feeding Buffer protects the Critical Chain
  - The Capacity Buffer protects the critical resources in a multi-project environment
<table>
<thead>
<tr>
<th>Technique</th>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analogy</td>
<td>Experience Based</td>
<td>Standard project?</td>
</tr>
<tr>
<td>Expert judgement</td>
<td>Fast, compensates for circumstances</td>
<td>Biases, incomplete recall of experiences</td>
</tr>
<tr>
<td>Delphi</td>
<td>Formal, Can use a mix of methods</td>
<td>As good as the experts</td>
</tr>
<tr>
<td>Algorithmic Models</td>
<td>Objective, Efficient, Calibrated</td>
<td>Subjective inputs, Inflexible, Must be calibrated</td>
</tr>
<tr>
<td>Critical Chain</td>
<td>Reduced time schedule</td>
<td>Extra pressure (?)</td>
</tr>
<tr>
<td>Lichtenberg</td>
<td>Top-down view, Risk Focus</td>
<td>Mentally demanding</td>
</tr>
<tr>
<td>Paired Comparison</td>
<td>Suggests high fidelity. Practical if comparisons can be reduced.</td>
<td>Cumbersome? Unproven in academia</td>
</tr>
</tbody>
</table>

Real world case 3

- Estimated effort for each use case during Inception:
  - Requirements
  - Design
  - Implementation
  - Test
- Allocated use cases to iterations
- Experiences – Sufficiently good. Precision mostly OK, large deviations (up to +100%) sometimes due to:
  - New architecture
  - Use cases with complex business rules
  - Most deviations struck execution of Elaboration phase – iteration 1
- Caution: Early estimates may turn out to be completely wrong if elaboration fails to stabilize architecture
A sketch of a practical method for *Inception*

- Order use cases with *paired comparison* according to effort
- *Detail a few use cases*, and analyze them in detail – estimate with Lichtenberg
- Use paired comparison to *estimate rest of use cases*
- Estimate *impact of mechanisms, risks and missing use cases* with Lichtenberg
- Make an iterative plan for use case and mechanism realizing
- Make schedule based on plan
- *No commitment yet!*

Tids- och kostnadsuppskattning av RUP Projekt: #23
RAIN 2002

A sketch of a practical method for *Elaboration*

- Learn from *Inception time spent vs plan*
- Improve estimates when there are *use case realizations available*
- Adjust to current knowledge of risks
- Study trends in growth in number of use cases
- Let the developers do the estimates!
- Manage the trend estimates

+ 
- Missing use cases: + 10%
- Non-functional requirements: 690h
- Mechanisms: 1550h
- Technical risk impact: 1300h
- Non-technical risk impact: 1200h

Tids- och kostnadsuppskattning av RUP Projekt: #24
RAIN 2002
When to use different techniques?

- **Function Points**
- **Use Case Points**
- **COCOMO 2**
- **Subsystem based**
- **Use case realization based**
- **Use case based**
- **Lichtenberg**
- **Paired Comparison**

**System analogy**

- TG0: Inception
- TG1: Elaboration
- TG2: Construction
- TG4: Transition

**Base technique!**

Manage the trends!

Tids- och kostnadsuppskattning
av RUP Projekt: # 25
RAIN 2002
**Next steps**

- Start using one technique
- Use it from day one in your project
- Let the project members do the estimation
- Collect experiences and improve
- Call us if you need more
- Thanks for your time!

**References**

- Steen Lichtenberg, Projektplanlægning i en foranderlig verden, Polyteknisk Forlag, Lyngby, 1978
- Goldratt, Critical Chain