Towards Automated Software Project Planning

Extending Palladio for the Simulation of Software Processes

Oliver Hummel & Robert Heinrich

KPD Symposium 2013
Position Paper
Motivation

FISCUS

“In it is very difficult to make a vigorous, plausible, and job-risking defense of an estimate that is derived by no quantitative method, supported by little data, and certified chiefly by the hunches of the managers.”

Fred Brooks [Wikipedia]
Overview

1. Problem: Software project planning is difficult

2. Idea: Simulate the development process

3. Benefit

4. Previous work

5. Proposed approach and necessary next steps
Problem

- Software projects are still planned like some twenty years ago
  - using relatively primitive techniques like bar charts or activity networks

- Effort is estimated with the help of parametric models such as COCOMO
  - based on relatively coarse-grained parameters and assumptions

  \[ \text{Effort} = A \times \text{Size}^E \times C \]

- No possibility to model concrete tasks, roles and their dependencies
Idea

- A look over the fence of software engineering reveals that other disciplines simulate their production processes

→ Can't we simulate software development processes, too?
Expected Benefit

- Simulating the software development process will allow to –
  - derive better project plans (automatically)
    - thus, helps to educate project managers
    - and allows a better project controlling
  - better understand the dependencies within a project
    - and hence reduce the risks of overlooking them
  - simulate various project flows in terms of –
    - selected development process
      - e.g. agile vs. waterfall
    - personell
      - where and when does it make sense to add new staff

➔ In a nutshell: it will help to minimize planning risks
Previous Work

- ... consists mainly of estimation tools that –
  - are based on COCOMO or similar models
  - derive probability distributions for schedule and/or effort
  - activities and their dependencies can’t be modelled

- The group of Ludewig (U Stuttgart) has developed a project management simulation for software projects (SESAM)
  - requires several hours of playing time

- To our knowledge: no tool that allows e.g. the comparison of different process models
  - or the effects of a delayed activity
Proposed Approach

- Based on a recent extension of PCM…
  - allowing to represent human actors and activities in a business process
- …the ambitious long-term vision is a tool that –
  - creates an optimized project plan for a given set of requirements automatically
    - based upon a selected process model and available resources
  - allow to manipulate individual tasks and assignments
    - in order to see their influence on the project outcome
- …the medium-term approach aims on –
  1. using (an extended) KAMP to derive a work breakdown structure (WBS) for an architectural model
  2. deriving a project plan and a Palladio model for the WBS
    - based on some desired process model
  3. simulating the derived model(s) in order to compare them
Necessary Extensions

1. Extend the PCM+BPM with missing concepts –
   - Artefacts resp. Documents
     - in the business process model
   - Task Types (as identified by F. Brooks)
     - extend simulation behaviour with appropriate traversal strategies
   - Teams
     - add a layer to the organization environment model
   - Deadlines
     - add an automatic detection of deadline violations

2. Extend and adapt KAMP
   - in order to allow the creation of a full WBS

3. Come up with a user-friendly user interface
   - including easy to use templates for SE process models
Summary

Problem
- Software projects are still planned like some twenty years ago.
  - using relatively primitive techniques like bar charts or activity networks.

- Effort is estimated with the help of parametric models such as COCOMO.
  - based on relatively coarse-grained parameters and assumptions.

  \[ \text{Effort} = A \times \text{Size}^a + C \]

- No possibility to model concrete tasks, roles and their dependencies.

Idea
- A look over the fence of software engineering reveals that other disciplines simulate their production processes.

Previous Work
- ... consists mainly of estimation tools that...
  - ...are based on COCOMO or similar models.
  - ...derive probability distributions for schedule and/or effort.
  - ...activities and their dependencies can’t be modelled.

- The group of Ludewig (U Stuttgart) has developed a project management simulation for software projects (SESAM).
  - ...requires several hours of playing time.

- To our knowledge, no tool that allows e.g. the comparison of different process models.

Proposed Approach
- The ambitious long-term vision is a tool that...
  - creates an optimized project plan for a given set of requirements automatically.
  - based on a selected process model and available resources.
  - allows to manipulate individual tasks and assignments.
  - in order to see their influence on the project outcome.

- The medium-term approach aims on...
  1. using an extended KAMP to derive a work breakdown structure (WBS) for an architectural model.
  2. deriving a project plan and a Palladio model for the WBS.
  3. based on some desired process model.
  4. simulating the derived model(s) in order to compare them.

TODOs
1. Extend the PCM+BPM with missing concepts –
   - Artefacts resp. Documents
     - in the business process model.
   - TaskTypes (as identified by P. Brooks)
     - extend simulation behaviour with appropriate traversal strategies.
   - Teams
     - add a layer to the organization environment model.
   - Deadlines
     - add an automatic detection of deadline violations.

2. Extend and adapt KAMP
   - in order to allow the creation of a full WBS.

3. Come up with a user-friendly user interface
   - including easy to use templates for SE process models.
Thank you for your attention!

Time for some questions…
  … or maybe you want to contact me later