Software Architecture, Process and Management

Project Management

Allan Clark

School of Informatics
University of Edinburgh

http://www.inf.ed.ac.uk/teaching/courses/sapm
Semester Two 2012-13
Clarifications — Everyone

- “Dear Allan” is fine, in particular I’m not a professor, but thanks for the implied promotion
- Your self-imposed deadline, also imposes a deadline on your feedback
  - Two weeks after your deadline I should give you adequate feedback
  - This is not so interesting to the level 11s who have one shot
  - Level 10s, you could potentially write a follow up post (at least for your second post) taking into account your feedback.
  - It will have to be about a different topic though
Clarifications — Level 11s

- Your only default deadline is Thursday 28th March 4pm, for submission of your report
- You may:
  - Impose an earlier deadline upon yourself by emailing it to me,
    - I suggest doing this by the end of this week, but that is only a suggestion
  - Email me proposing a topic, which I will approve (or reject) and may generally comment upon.
  - Note though that I only guarantee to respond within three working days
- You may submit before your deadline, but:
  - you may re-submit before your deadline
  - Hence I won’t start to mark it until your deadline
  - The deadline for me giving returning your feedback and provisional marks is two weeks after your deadline, not when you submit it.
  - If you intend it to be your absolute last version let me know.
Software and Buildings

- In the first lecture I compared software projects with building projects.
- Nobody believes that one can take 100 people who can individually build a rabbit hutch, put them together, and obtain a team capable of building anything suitable for humans, even a small cottage.
- Yet this essentially the expectation for large software projects
  - Take 100 people who can individually create small scripts
  - Put them in a bunch of cubicles
  - You will have a new operating system in 3-4 months
- Nobody looks for the most successful brick-layer and asks that person to design a new large building
  - The person they look for is called an **architect**
First things first

- A large software project usually begins with the hiring of several software developers
- These are expected to start coding, more or less immediately
- A large building project does not begin in this way
- It begins with the hiring of an architect
  - The architect designs the building
  - Contractors bid for the contract to build the building
    - There is often some back-and-forth between the architect and the constructors at this point
  - The contractors then plan the construction
  - Eventually constructions begins
Differences

- Let's be fair there are some differences between software and building construction
- A software project is more malleable to change
  - If a given design isn’t working it can be changed
  - Requirements change, sometimes called “feature creep”
- Construction projects on the other hand cannot get halfway through only to realise that the design is not stable
- Customers often do not know what they want, they are not often software experts themselves
  - But everyone is fairly expert in building usage
Differences

- We have been building buildings for a lot longer than we have been building software
- This is a picture of Skara-Brae in the Orkney Islands
- It is *Neolithic* occupied around 3180 BCE-2500 BCE
Company Origins

- The world’s oldest known company
  - Kongō Gumi Co., Ltd
  - Japanese construction company founded in 578
  - Unfortunately went to administration in 2006
Company Origins

- The world’s oldest known company
  - Kongō Gumi Co., Ltd
  - Japanese construction company founded in 578
  - Unfortunately went to administration in 2006

- Not really true, as Microsoft was founding in 1975
Uhm

This is what happens when you establish your company in the 70s
Quick Pop Quiz

- When was the company Nintendo established?
- What kind of company is Nintendo?
Quick Pop Quiz

- September 1889
- It’s now a video game company but originally was a card company, then a toy company
Despite these differences, large software projects are nevertheless still large projects.

As such many successful approaches for managing large projects have been adopted for the management of large software projects.

Some successfully, some not so successfully.

In the remainder of this lecture we will look at some of these approaches

and attempt to relate them to software projects

This lecture though is mostly to set the basis for the discussion in future lectures.
Project Management

- A project is ‘a temporary endeavour to produce a unique product, service, or result’ (PMI 2004).
- Project management techniques were originally developed for engineering projects like building construction.
- Project management focuses on planning, scheduling, monitoring, and controlling the complex interdependencies among subtasks.
- Project management techniques are most relevant when considering the entire project in which software development is embedded, which includes other activities such as documentation, training, hardware, etc.
Example Project

Consider building a garden shed, which involves:

1. Designing the shed
2. Figuring out what materials are needed
3. Ordering or purchasing the materials
4. Putting together the various parts

Some of these tasks depend on the others, some must be scheduled, some take labor, etc.
Project Management Tasks

- Probably before work starts, need to figure out:
  - What needs to be done
  - What order they can be done
  - How long each will take
  - How long the whole project must take in principle
  - How long the whole project is expected to take, given finite resources
Project Scope Statement

- In order to make any decisions about the project, it needs to be made concrete.
- This is done by writing a scope statement, such as:
  - We will build a garden shed capable of holding 2 bicycles, a lawn mower, and a small workbench
  - Planning permission will not be needed
  - We will not connect it to electric or water mains
  - It cannot cost more than 1000 pounds or take longer than 1 month (may be a constraint rather than scope)
- Note that the scope statement includes not only what it will do, but what will not be included.
Work Breakdown Structure

- A WBS is a diagram showing the major subtasks of the project:
- Rule of thumb: break things down as far as necessary to estimate and schedule them, and no further.

```
Build Shed
  ├── Planning
  │     └── Obtain Materials
  │        ├── Construction
  │        │     ├── Pour foundation
  │        │     │     └── Finish (door, etc)
  │     └── Build frame
  │        └── Install siding
  │            └── Install roof
  └── Install roof
```

Pour foundation
Build frame
Install siding
Install roof
Finish (door, etc)
Network diagrams can be constructed from the WBS, adding dependencies and estimated durations.

Note that you can have different network diagrams for the same project, depending on your assumptions and approach.
The critical path is the longest path through the network diagram.

It is the minimum duration of the project if there are infinite resources (so that tasks can occur in parallel) and accurate estimates.

Here, everything but putting up the siding is on the critical path, and must happen in the order specified.
Tasks not on the critical path have *slack*.

- the duration by which they can be late without making the project later than the critical path duration.
- Here, putting up the siding has one hour of slack: one hour longer than estimated is ok.
- but if it takes longer it will be the new critical path, and delay the project.
- Recall, this is assuming you have enough resources (in this case, people and tools) to allow these two jobs to proceed in parallel.
FAPM — Formal Project Management

PERT/CPM Charts

- Network diagrams come in a variety of flavors with different names:
  - A Program Evaluation and Review Technique (PERT) chart shows dependencies and time estimates, using 3-point estimates.
  - The Critical Path Method (CPM) chart is a related alternative, using single estimates.
- Both show similar information, but use different methods for calculating the critical path and slack.
A Gantt chart shows the tasks and their durations graphically, in calendar form, with one bar per activity.

The bar shows the earliest start date and expected duration.
Gantt charts can also show

- Summary activities (not here)
- dependencies with lines (see SBML to Bio-PEPA)
- Milestones, the dotted crosshairs

The dependencies make slack clear (see SBML to Bio-PEPA)
How do we decide how long each task will take?

- Mostly by guessing
- Many companies keep data on estimates
How do we decide how long each task will take?

- Mostly by guessing
- Many companies keep data on estimates
- Let’s play a quick game

The task:
- Write a Java to HTML translator
- It needn’t do any error detection/correction
- But nor need it translate invalid JAVA
- You cannot use an existing one, but you can use libraries for parsing text or even parsing Java itself.
How do we decide how long each task will take?

▶ Mostly by guessing
▶ Many companies keep data on estimates
▶ Let’s play a quick game
▶ The task:
  ▶ Write a Java to HTML translator
  ▶ It needn’t do any error detection/correction
  ▶ But nor need it translate invalid JAVA
  ▶ You cannot use an existing one, but you can use libraries for parsing text or even parsing Java itself.
How do we decide how long each task will take?

- Mostly by guessing
- Many companies keep data on estimates
- Let’s play a quick game
- The task:
  - Write a Java to HTML translator
  - It needn’t do any error detection/correction
  - But nor need it translate invalid JAVA
  - You cannot use an existing one, but you can use libraries for parsing text or even parsing Java itself.

- Write down, how long you think this would take a single developer?
SAPM: Estimating Tasks

How do we decide how long each task will take?

▶ “Planning Poker” is a game doing this rounds in development circles

▶ There are cards for: http://www.crisp.se/bocker-och-produkter/planning-poker

▶ And a web-based implementation: http://www.planningpoker.com/
Resources

▶ All these charts assume infinite resources
  ▶ things that can logically happen in parallel, are assumed to happen in parallel.

▶ In reality, there are limited resources, so there are often many dependencies not shown in the charts.

▶ E.g. if only one person is available to work on the shed project, the siding and roofing tasks must be done in serial, due to the resource constraint.
Crashing/Fast Tracking

- If the critical path is still not fast enough, its possible to shorten the duration by changing some assumptions.

- **Crashing:** Change the duration of some critical task, e.g., if it is possible to parallelize it by assigning more people to it.

- **Fast tracking:** Allow tasks to be done in parallel by changing the logic in the network diagram.

- If tasks can be parallelized (as in building construction) these allow total duration to be greatly reduced.
SAPM — Formal Project Management

For More Information

- Project Management Institute (PMI): Largest international project management organization; US-based (pmi.org)
- International Project Management Association (IPMA): Oldest international project management organisation (www.ipma.ch)
- Association for Project Management (APM): UK-based national organisation (www.apm.org.uk)
- Project Management Glossary: www.maxwideman.com/pmglossary/
Use Project Management for your Software Projects?

- **Why?**
  - Software is just a project like any other, with tasks, interdependencies, resources, etc. Right?

- **Why not?**
  - The vast majority of software development tasks cannot be estimated up front
  - Doing so requires knowing the requirements and the design, plus how long debugging will take
  - After those three tasks there’s very little work left.

- **Clear win for Project Management:**
  - Software projects embedded into larger projects.
  - Here, the software must be delivered on time for the rest of the project to succeed, and depends on the other parts. In such cases, project management is very useful.
  - Within the software-only portion, perhaps not.
Summary

- Project management charts and techniques are useful for scheduling tasks with complex dependencies
- WBS: main tasks
- Network diagram: dependencies, durations
- Gantt: calendar
- Be careful not to trust the charts just because they look fancy; e.g., many constraints are not shown
SAPM — Related Reading

- **Required Reading**
Suggested Reading

- ACM/IEEE Software Engineering Code of Ethics
Any Questions?