Conquest 2006

Planning Software Development:
Taken as a manufacturing control problem

DI. Andreas Nehfort

andreas@nehfort.at www.nehfort.at

Agenda

- The established paradigm: project management
- What is a project?
- An example to illustrate the problem
- How does project management work?
- How does manufacturing control work?
- Manufacturing optimising targets and strategies
- Some approved rules for planning SW-development based on the concepts of manufacturing control

The established paradigm

Software Development

- **Project**
  - **Project Management**
    - **Project Planning**

Planning Software Development = Project Planning

All models are wrong, but some are useful

For the planning of many SW-development problems

- this project management approach is **helpful**
- but for some it is **obstructive**!

Many SW-development tasks have more in common

- with the order disposition in a flexible small lot production
- than with a project!

Then project oriented planning gets complicated, tricky and cumbersome.

What is a project?

"A project is a temporary endeavor undertaken to create a unique product or service."

"**Temporary** means that every project has a definite beginning & a definite end." "The end is reached when the **project objectives** have been achieved, or … the project is terminated."

"**Unique** means that the product or service is different in some distinguishing way from all other products or services."

"For many organizations projects are a means to respond to those requests that **cannot be addressed within the organization's normal operational limits**."

PMI - PMBOK 2000

Project Objektives

"The **objectives** of **projects** and **operations** are fundamentally different:

- The objective of a **project** is to attain the objective and **close the project**.
- The objective of an **ongoing non-projectised operation** is normally to **sustain the business**.

Projects are fundamentally different

- because **the project ceases** when its declared objectives have been attained,
- while non-project undertakings **adopt a new set of objectives and continue to work**."

PMI - PMBOK 2000

An imaginary project
The starting point

Useful SW Inc. started the development of UseMe 3.1:

- Project start: January
- Deadline: End of Sept. (before the large exhibition in Oct.)

The planning positions:

- Request of the product management: 12 – 14 new features
- Position of the project manager: 10 new features at most!

Agreed function size for Rel. 3.1:

- **10 new features**

The project evolves ...

Unfortunately the project is interfered by some incidents:

- February 15th: An urgent bug-fix (and patch) for Rel 3.0 is required.
  The analysis shows: The problem is trickier than expected …
- March 2nd: Mrs. Meier – who leads the UseMe-implementation project at TheBigOilCompany needs a new feature until April 10th to fulfil the contract …
- March 20th: The product management has a change request:
  A new feature is required to respond to a competitors product announcement (in exchange feature 3 could be postponed – if absolutely necessary) …

The project evolves ...
→ change management

- In April the developers find out that feature 6 cannot be implemented as planned: “We need more time to find an adequate solution.”

And so on:

- Each month one or two major change requests
- and lots of minor ones …

The change requests are handled by the change management and the change control board.

The result

At the end of September the project team has implemented:

- 4 of the initially 10 planned features,
- plus 2 features proposed by the product manager during development,
- plus 3 features needed by customer projects,
- plus 2 patches for Rel. 3.0:
  - The urgent patch from March
  - and a „service pack“ in July.

The result ⇔ Planning

The output:

- 11 features and patches instead of 10.

But:

- Only 4 features from the initial plan for Rel. 3.1.
- 6 for Rel. 3.1 planned features were **not** implemented!

The overall performance seems not so bad!

But:

- Nobody was satisfied with the result!

The cost overrun amounted to approximately 25%

- But - that was not the problem …

The perspective of the project team

The project management and the developers were under pressure from February 15th to September 30th:

- Working overtime was a matter of course,
- 60 working hours per week were no exception.

The main topics at the weekly status meetings:

- What has NOT been done?
- Why could it not be done? → Justifications
- Trouble shooting and new change requests
- Replanning

The high employment of the team was not appreciated!

The perspective of the management

The product management and the development management were not satisfied with the **poor development performance**!

- Only 4 of the 10 planned features were implemented! - OK 6!
- Thereby we wanted 12 to 14 new features for Rel. 3.1!

Rel. 3.1 was announced with 10 new features in **February**:

- The customers asked for the missing features at the exhibition in October - It was a hard time for the exhibition team.

The 4 features for customer projects and the 2 patches **are NO argument**! If we are not even able to manage them, we might as well close the company!

Rel. 3.1 from the customer's perspective

Most of the customers were disappointed about Rel. 3.1!

Many customers had been waiting for the new features.

- Some of them wanted to upgrade to Rel. 3.1 in the last quarter of the year.
- Some had left out Rel. 3.0 for this reason.

Many customers felt insecure – that was not the first time the product did not comply with the announcement!

Your opinion of the project?

Does this story seem totally overstated to you?

- …

Or do you see similarities to real world projects?

- Have you ever heard about such a project?
- Does the story sound familiar to you?
- Have you even worked in such a project?

My analysis

From my point of view the development of Rel. 3.1 is NOT a project!

The project objective:

- Can be changed optionally!

The time limit → definite end:

- Concerning the deadline: **YES – there is a definite end!**
- Concerning the (initial) objective: **NO evidence for a definite end!**

What is our project team doing in the beginning of October?

- Closing the project?
- **NO!** ... They take the next 10 features and start the development of Rel. 3.2 → **„ongoing work“**

The basic idea behind a project

We need **the whole thing** → we have to plan the whole thing!

- We cannot treat requirements/features isolated
- We cannot remove or add single features

(I know - that is a black/white snapshot)

In our example project this precondition is not given:

- We can treat a lot of requirements/features isolated!
- We can remove or add single features!

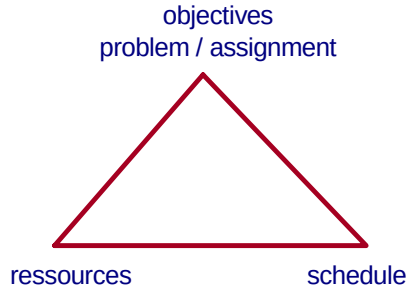
How does project planning work?

The baseline for planning: The problem/assignment

- It is derived from the project objectives
- It represents the amount of work.
- It is the **independent planning variable**!

The dependent variables are:

- **Effort** respectively **costs** and
- **Time (schedules)**



Adapting the project plan

The primary approach:

- **The use of resources** (→effort) and **schedules**
  - should be **adapted** in a way,
  - that the **project objectives** will be achieved.

The fallback solution:

- When the primary approach does not work (we cannot achieve the objectives within the limits)
  - we have to adapt the problem/assignment
  - and **re-plan** the **re-dimensioned project**.

Development and production planning

Software development as our example has a lot in common with **production disposition** → **flexible order production**:

- The feature → a production order
- The list of open features → the order backlog

Agile development works with these metaphors and offers adequate planning techniques!

"Assembly-line production techniques apply to software, too"

Mary Poppendieck, 2001 - Lean SW-Development

How does production planning work?

The baseline for planning:

- The **available resources** (per time period)

The **resources** are the **independent** planning variables!

The **dependent planning variables** are:

- The **resource allocation** → **Order** and **time**
- The **overall performance** → the **output**

Adapting the production plan

The primary approach:

- **Orders are assigned to the available resources**:
  - **Resource** ← **Order** and **time**
- For this the order backlog is:
  - **Prioritised** → Schedule
  - **Categorised** → Needed resources

The fallback solution:

- When the primary approach is not sufficient
  - we have to adapt the available resources
  - and **re-plan** with the **redimensioned resources**.

Production planning objectives

Production planning (disposition) can have different targets:

- Optimising throughput → **maximum productivity**
- Optimising throughput time → **short-term delivery**

It is a matter of fact regarding manufacturing control that

- you cannot optimise in both dimensions simultaneously,
- without making compromises!

Further optimisation goals:

- Optimised use of bottleneck resources
- Minimising semi-finished products (fixed capital)
- Minimising production costs, …

Optimising throughput

Main target:

- Optimising throughput by minimising non-productive times:
  - Set-up times
  - Downtimes, re-tool times

Measures for optimising throughput:

- Appropriate order sequence → Set-up optimisation
- Secured parts supply

Optimising throughput time

Main target:

- Minimising the time between order and delivery:
  - The duration before start of production
  - The wait times between the separate working steps

Measures for optimising throughput time:

- Reduction of order backlogs in the production
- Pull from demand / Canban control
- No production start without reserved bottleneck resources

Optimising bottleneck resources

Main target:

- **Optimising throughput of bottleneck resources by:**
  - Set-up optimisation
  - Avoiding downtimes

Measures for optimising bottleneck resources:

- Optimising the order sequence to bottlenecks
- Order buffer before bottlenecks
- Secured parts supply for bottleneck resources

Minimising semi-finished products

- Semi-finished products represent fixed capital, which is bound to not yet clearable orders.

Main targets:

- Minimising the **fixed capital**
- Minimising the **quantity** of semi-finished products
- Minimising the **duration** of the semi-finished status

Measures for minimising the semi-finished products:

- In the parts store → Just-in-time supply
- In the production → small stocks, short throughput-times

Minimising production costs

Main target:

- Maximising the **margin & profit**!

Measures for minimising production costs:

- Minimising the staff contingent → Automation
- Minimising wage costs → Offshoring
- Maximising the machinery utilisation → Shift operation
- Minimising material costs
- Minimising logistic costs, …

Depending on the cost structure of the respective product.

Optimising in SW development

Which optimising target do we have in SW development?

- To be on schedule?
- Project effort / costs?

How consequent do we strive for our optimising targets?

- Are the optimising targets clear?
- Are we aware of alternatives?

Have we got clear optimising strategies?

- Are we aware of the interdependencies?
- What are the adverse effects of optimising one parameter?

Planning Software Development

Taken as a manufacturing control problem

When should we apply the concepts of manufacturing control to software development?

- When the available resources per time period are constant.
- When the result cannot or should not be fixed in advance!

How can we apply the concepts of manufacturing control to Software Development?

- Let‘s use the analogy to manufacturing control as inspiration!
- And ... I can provide some rules ...

Some approved rules

Segmentation into planning horizons:

- Long-, mid-, short-term & ad-hoc

Strict prioritisation of the tasks:

- XP-planning game

Segmentation of the resources:

- The 50% rule
- The 1/3 rule

Segmentation into planning horizons

Long-term → Future releases:

- Typical planning horizon: > 1 Year

Mid-term → Development of the next release:

- Typical planning horizon: 9 – 12 months

Short-term → Urgent change requests :

- Typical planning horizon: a few weeks to months

Ad-hoc → urgent bug-fixes:

- Typical planning horizon: a few days to weeks

The consequence: Segmentation of the resources

Segmentation of resources: The 50% rule

Planning approach:

- 50% of the resources for PRIO-1-features
- 25% of the resources for PRIO-2-features
- The rest (25%) remains for requests & (estimation-) errors

Best-Case-Result:

- All Prio-1- and Prio-2-features are implemented in time (if the interferences add up to less than 33% of Prio-1)

Worst-Case-Result:

- All Prio-1-features are implemented in time (if the interferences add up to less than 100% of Prio-1)

Segmentation of resources: The 1/3 rule

Planning approach:

- 1/3 of the resources for PRIO-1-features
- 1/3 of the resources for PRIO-2-features
- The rest (1/3) remains for requests & (estimation-) errors

Best-Case-Result:

- All Prio-1- and Prio-2-features are implemented in time (if the interferences add up to less than 50% of Prio-1)

Worst-Case-Result:

- All Prio-1-features are implemented in time (if the interferences add up to less than 200% Prio-1)

Prioritisation of the tasks
XP-planning game

Basic rule:

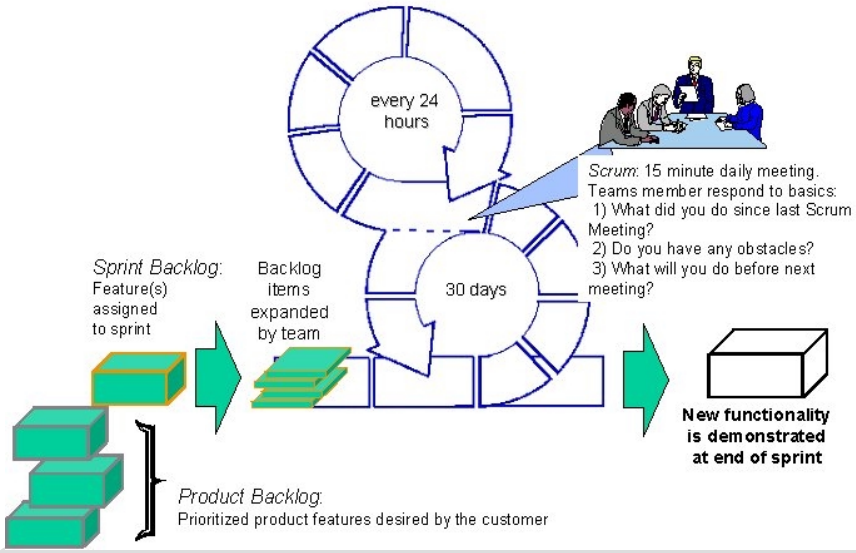
- The customer makes the „Business Decisions“ → Priorities
- The supplier makes the „Technical Decisions“ → Effort

For each iteration: Agreement for the scope of delivery:

- Planning base line: The available resources
- Planning input: The list of features (with priorities)
- The control parameters: Feature points, project velocity, …

During iteration the planning is not to be changed!

SCRUM ➔ 30 day sprints



All models are wrong, but some are useful

Which model is helpful for your planning problems?

- The **project paradigm**?
- The **manufacturing control paradigm**?

Which optimising targets are relevant for your planning?

- …

When we choose the manufacturing control paradigm:

- We have to redefine our planning problem!
- We change our point of view!

Thank you for your attention!

Questions & Discussion