Chapters 10 and 11

Overview

The Theory of Constraints

- Any system will have one constraint that is limiting its throughput.
- To make things better, we have to find that one constraint and
  - increase its capacity
  - offload the work onto other parts of the process
  - eliminate the constraint entirely

Three step example - Baking Cookies

- Mixing the dough
- Cooking the cookies
- Icing the cookies

Ponder:

- Where's the bottle neck?
- How can we reduce the bottleneck?
- How does the type of icing in our orders drive production?

Planning Goals

- Maximize *return on investment* of our iteration
  - Build most valuable functionality
  - As quickly as possible
  - Paying attention to design strategies that reduce risk
- Fail early if we are going to fail
  - Don't just pick *low hanging fruit* if there are risky parts left to build.

Royce

What did Royce do to try to encourage failing early? (he didn't call it that, but he was definitely worried about failing late)

Planning Artifacts

Story One or two sentences that describe a piece of functionality from a user's perspective

Product Backlog A prioritized list of stories

Product Owner The person who is responsible for maintaining the product backlog

Planning Game

- Players
  - Customer
  - Development
- Pieces: Story Cards
  - date
  - type of activity: new, fix, enhancement, or test
  - priority (user and development)
  - description
  - risk
  - estimate

Story details

The author lists a bunch of other things that can go on a story card. When would each of those be valuable?

Phase 1: Exploration

Legal Moves:

- Write a story (customer)
- Estimate a story (development)
- Split a story (either)
  - Too big to estimate
  - Parts have different priorities

Example

Make a couple of example stories for your project.

Phase 2: Commitment

- Assign risk (development)
  1. can be estimated precisely
  2. can estimate with reasonable confidence
  3. cannot estimate
- sort by value (business) into three piles
  - necessary for the system to function
  - less essential, but of significant business value
  - nice to have
- set velocity (development) (how fast the team can program in an appropriate metric/calendar month)
- choose scope (business)
  - must select stories so they fit in the project velocity

Phase 3: Steering

- scale back velocity (development)
  - can ask business to narrow stories in the current release
- new story (business) in mid-release
  - can add a new story if it removes undeveloped stories of equal estimates
- reestimate (development)
  - if plan is no longer accurate, development can reestimate all undeveloped stories and reset the velocity.

Steering

What kinds of events could cause us to need to steer?

The Iteration Planning Game

Similar to the Planning Game, but used to plan the individual tasks/assignments that will build the functionality committed to in one iteration.

- Pieces - Task Cards
  - Description of something we need to do/build
- Exploration Phase Moves
  - turn the stories into tasks (create task)
  - split a task/combine tasks - want each task to be estimated to be between 0.5 to 3 days

The Iteration Planning Game

- Commitment Phase Moves
  - accept a task (by an individual)
  - estimate a task (responsible individual)
    - can be conditional on getting help from someone else
    - ignore pair programming - it shows up in calculation of project velocity that also accounts for meetings etc.
  - set load factors (each individual)
    - percentage of time you will spend actually developing
    - comes from historic metrics (never higher than 0.5 and rarely higher than 0.3), but can account for planned vacation/holidays
  - balancing (all)
    - add up your time and make sure no one is overcommitted

Differences from Planning Game

- Individual accepts a task before estimating it
- There may be tasks that are not related to needs of the customer (tool development, major refactoring, etc.)

Accepted Responsibility

How do you think this plays out? Do you always get the tasks you want? Are you pigeon-holed into one type of task? How do you expand your experiences?

Individual Estimates

How do we handle the peer pressure of individual estimates? Am I a weaker engineer because it takes me longer to do a particular task? Remember that "Courage" value . . .

Units of Estimation

- For Stories: Story Points
  - Measure of effort - not time! Remember: time depends on the individual
  - Relative - not absolute
  - We'll make the smallest story worth one story point
- For Tasks: Ideal Engineer Hours
  - How long it takes if everything goes perfectly.
  - Risk: pointy haired boss . . .