Living with Dwarfs

It’s common to have product backlogs influenced by others, but sometimes our backlogs are outright hijacked. In the spirit of Disney, we’ve dubbed this “Living with Dwarfs”.

Instructions

- Walk around and read the placard at each table. If you answer a question yes, pick up a dwarf.
- Once you have visited all seven dwarfs, stop and grab a seat.
Agenda

- Opening exercise: Living with Dwarfs
- Stay or go
- Introductions
- Product Owner and release plans
- Simulation: How do I plan beyond the current release?
- Closing

Stay or Go...

Consider going if...

- You have successfully planned long term with agile
- You don’t need to plan long term (small company?)
- You don’t want to do hands-on-work

We won’t be offended if you leave!

Other sessions in our track include...

- Taking Small Steps in a Big Organisation (An experience report on implementing ET)
  (John Stevenson)
- The Impact of Agile Quantified: A de-Mystery Thriller (with new metrics and data for 2014)
  (Larry Maccherone)
Stay or Go…

**Consider staying if…**

- Work on projects where plans change often
- Want to learn the basics about high speed estimation with agile
- Want to get the basics of what release planning should look like in agile

---

**Introductions**

Us
You
Donald Patti

Director of Applications, AAMC
+1.866-478-8080
+1.202-910-0245
dpatti@aamc.org
www.aamc.org
www.linkedin.com/in/delplayadon/

David Bulkin

VP of Training Services LitheSpeed
- +1.215-764-6822
- David.Bulkin@lithespeed.com
- @davidbulkin
- www.LitheSpeed.com
- www.linkedin.com/in/davidbulkin
Product Owner & Release Plans

Living with Dwarfs

- The Product Owner, owns the product backlog and the release plans.
- The Product Owner has to balance competing objectives.
Every Stakeholder Wants Something
The C-Level

C-Level Executives want to control the roadmap and vision.

Roadmap
Business Model
Revenue Model
Budget
Constraint Stories
User Experience
Usability Testing

Vision
Backlog Grooming
Features
User Stories
Market Research
Personas
Focus Groups

CEO

Every Stakeholder Wants Something
The C-Level

Finance wants to control the revenue model and budget.

Roadmap
Business Model
Revenue Model
Budget
Constraint Stories
User Experience
Usability Testing

Vision
Backlog Grooming
Features
User Stories
Market Research
Personas
Focus Groups

Finance
Every Stakeholder Wants Something

**Summary**

**C-Levels:** company strategy, pet projects
- Provide: funding to hit financial targets
- Require: long plans & progress against them

**Product Manager, Information Owner:** product / system strategy
- Provide: product roadmap & revenue model
- Require: new product features that align with organization’s strategy and roadmap

**Sales:** meet needs of specific prospects
- Provide: input on customer needs
- Require: features on the roadmap, or even in current sprint to land a specific customer

**Other Teams & Vendors:** functionality
- Provide: functionality we need, APIs
- Require: functionality they need, APIs

**IT Architecture & Infrastructure:** architectural runway
- Provide: runway of infrastructure to build against & standards to follow
- Require: long term notice for something different than company standard

**Legal & Compliance:** stay in business
- Provide: non negotiable rules & deadlines
- Require: work to meet compliance requirements and proof thereof

**Marketing:** company image, target audience
- Provide: branding standards, market research
- Require: lead time for request, advertising & promotional information

**Finance:** allocates funds
- Provide: budget and funds
- Require: updates on funds spent

**Others:**
- Security
- Privacy

---

A Release Planning System

Everyone wants something different, so we need an **objective system** for making release planning decisions and a process that supports continual updates to the plan.
Release Planning Basic
Iterative & Incremental Backlogs

Iteratively Define, Incrementally Deliver

Iterate your ideas and incrementally deliver

Image courtesy of Phil Bryant Alert Logic and Vincent Van Gogh
Iterative and Incremental

Which apply when we say we are iterative and incremental?

✔ We start with a credible, but not complete idea.
✔ We release small, frequent, chunks.
✔ We use what we learn from our frequent releases to refine our goals and plans.

- Start with a credible, long term plan.
- Iteratively and continually update the plan based on learnings and feedback.
### Roadmap to Backlogs

#### Roadmap
Plan features delivery and architecture updates for customer segments while accounting for legislation, industry events etc.

#### Product Backlog
List of things to deliver over long periods. Contains one or more release backlogs.

#### Release Backlog
List of things to deliver for a release. A subset of the product backlog.

#### Sprint Backlog
List of deliverables and tasks for a sprint (often two weeks).

---

#### High Speed Estimating

**The Fruit Bowl Example**
At release planning some teams use feature points, others go directly to story points. Either way, use relative measures; not days or hours.

### Relative Size

#### Table

<table>
<thead>
<tr>
<th>ID</th>
<th>PB Item</th>
<th>Priority</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>182</td>
<td>User Story</td>
<td>High</td>
<td>2</td>
</tr>
<tr>
<td>218</td>
<td>PBI</td>
<td>High</td>
<td>3</td>
</tr>
<tr>
<td>045</td>
<td>User Story</td>
<td>High</td>
<td>8</td>
</tr>
<tr>
<td>098</td>
<td>User Story</td>
<td>Med</td>
<td>2</td>
</tr>
<tr>
<td>076</td>
<td>User Story</td>
<td>Med</td>
<td>8</td>
</tr>
<tr>
<td>089</td>
<td>User Story</td>
<td>Med</td>
<td>13</td>
</tr>
<tr>
<td>186</td>
<td>User Story</td>
<td>Low</td>
<td>100</td>
</tr>
<tr>
<td>148</td>
<td>User Story</td>
<td>Low</td>
<td>100</td>
</tr>
</tbody>
</table>

#### Points Scale

- A common relative point scale is 1, 2, 3, 5, 8, 13, 20, 40, 100.
- This scale is as it seems.
  - A two is twice as big as a one.
  - A three is 50% bigger than a two.
  - Etc.

- Planning Poker is a popular approach for creating relative estimates, we will review a faster estimation game.
We are Judges in a Fruit Eating Contest

- We will be judges for a fruit eating contest.
- We will wash the fruit, ensure it is ripe, and make tools like knives and peelers available to participants.
- Participants will prepare and then consume the commonly edible portions (i.e. no stems, leaves, pits, etc.) of the fruit.

We Need to Estimate Relative Size of Fruits

- We need to estimate the relative effort that it will take the participants to prepare and eat each fruit (excluding leaves, stems pits, etc.).
- We have established a set of references fruits and will play a high speed estimating game for the remaining fruit.
### Fast, Whole Team, Estimates

Use “whole-team”, “high speed” approaches to quickly create estimates.

This allows the team or teams to quickly estimate new work so that planning can be continual.
Release Planning, When & Where

Release Planning and the Scrum Framework

Where is Release Planning done in the Scrum Framework?

Let’s see.
Release planning is part of the continual backlog refinement process, but most teams do just a bit of JIT release planning at Sprint Planning.
Relative Estimates at Sprint Planning

Create relative estimates to see how much they can commit to.

We can do these, they are 50 pts.

Release Planning at Sprint Planning

If we do relative estimating at Sprint Planning, and use the estimates to make our commitments we essentially have no ability to plan long term.

1. True
2. False
Release Planning at Sprint Planning

Having no long term plan can be a problem for some teams?
1. True
2. False

Release Planning and the Release Train

SAFe has popularized the Release Train concept, which is a quarterly look-a-head
Release Train
* A quarterly look-a-head

Teams get together quarterly, for a couple of days, to plan ahead for the next three months.

A Model of Quarterly Planning

At the quarterly session, a Program Backlog is decomposed into team backlogs, with a quarterly look-a-head.
An Idealized Model

- Quarterly Long Term Release Planning
  - Program Backlog
  - Team Backlog

- Continual Adaptive Planning
  - Strategic Planning (Product, Purpose)
  - Continual Release Refining (Mini Sessions)
  - Tactical Backlog Grooming

- Sprint
  - Sprint Planning
  - Daily Scrum
  - Work
  - Sprint Review
  - Sprint Retro
  - End of Sprint
  - "Done" Stories / Product Increment

- Sprint Backlog
- Burndowns

- Start of Sprint
- Daily

Quarterly Release Planning

- Columns are time boxes.
- Rows are individual work streams.

- Two day event for whole team and all dependents teams.
- Meet to do discovery, identify risk items, estimate and create a cross team plan.
Team and Cross Team Refinement Sessions

- Twice a week, post Scrum, each team spends up to 30-minutes to estimate new work and adjust plans.
- Later that day a subset of teams have a multi-team planning session to review progress and adjust across teams.
- Ad-hoc sessions happen as necessary.

<table>
<thead>
<tr>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad-Hoc</td>
<td>Cross Team Planning</td>
<td>Ad-Hoc</td>
<td>Cross Team Planning</td>
<td>Ad-Hoc</td>
</tr>
<tr>
<td>Team Release Planning</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Rolling Wave

Release Planning should be a continual rolling wave in a team and across teams.
Release Planning System with AppetizerZ

A Simulation

Business Value via MoSCoW

**MoSCoW** is one way to denote value. We used it for our simulation.

- **MUST** have.
- **SHOULD** have if at all possible (workarounds exist)
- **COULD** have if it does not affect anything else.
- **WOULD** like in the future but won’t have now.

SMART Sequencing

One viable model; created by David & Don.

1. Sort by priority and size, highest priority, smallest first.
2. Move items to account for dependencies.
3. Assign into schedule buckets accounting for velocity.
4. Realign (i.e. move) for external lead-time.
5. Transfer (i.e. move) items to adjust for deadline.

Simulation Overview

You are about to complete a simulation of release planning by creating and adjusting a product roadmap over a period of 1 ½ years – broken out into 6 quarters.

This concept applies at all levels from roadmap down to sprint, we are do this at a high level so that we have less cards to work with.
AppetizerZ

Our simulation is AppetizerZ, a mobile software product that brings like minded people together over food. Through the map-based software, people choose food and personal interests, then are matched with a combination of restaurants and people nearby who share their interests so they can meet-up.

AppetizerZ - The Startup

AppetizerZ is a start-up with four employees starting to develop their product. As their VP of Product Development and product owner, you are being asked to create the schedule for the feature list.

Assume your team has a velocity of 500 feature points per quarter.

Important! Be sure to consider external factors, lead-time, deadline and dependencies in your decision-making.
A Release Planning System

An objective system for release planning includes:

- Stakeholders determine value.
- Team(s) estimates size.
- The system determine sequence to market.

Closing
Predictagility

- Not everyone needs to plan long term.
- If you need to accommodate for longer lead times and dependencies then…
  - Make a long term plan (perhaps 18 months)
  - Acknowledge that plans will change
  - Make short term commitments (perhaps 6 months)
  - Revisit frequently (perhaps twice weekly)