Agile Release Planning and Monitoring
March 1, 2010

A Tough, but Solvable, Problem

Release Planning

Types of Release Plan
Our Problem
Function Point Discussion
Our Solution
Release Plan
**Planning Artifacts**

- **Project Visioning**
  - Product Vision
  - Price / Contract
  - Product Roadmap
- **Project Start**
- **Release Planning**
- **Sprint**
- **Release**
- **Closeout Activities**

**Sprint Goal**
- Sprint Backlog

**Release Goal**
- Release Baseline
- Release Strategy
- Release Game Plan

**Release Types: What’s the Problem?**

- When Can we Release?
  - Known scope, existing team
- What Can we Release?
  - Known time, existing team
- How much will it Cost?
  - Known time, known scope, new team
  - This is the tough one… fixed price
  - All based on calculation

- Time x Capacity = Scope

- Let’s do a quick version of the “hard” case

- Then finish the talk with one of the easy cases
  - Because I have the data for it 😊
Our Problem

- Have a client, Royal Catalina Airlines (RCA)
  - Royal Catalina Airlines is owned by Sir Geoffrey Smithers (SirJeff) who made a fortune writing software in the Silicon Valley before buying a plane and ferrying tourists up and down the California Coast
  - He is now buying 4 more planes, hiring pilots, crews, etc, and wants a website, RoyalCatalinaAir.com. We’re writing it for him…
- In six months he wants:
  - The customer can “Buy an e-ticket”
  - The customer can “Check Status of Flights”
  - The customer can “Get a Hotel at his destination”
  - The customer can “Get a rental car at his destination”
  - The Pilots can “Manage Pilot Timesheets”
  - The Customer can “Manage the Good Customer Plan”
- He asks: “How Much will this Cost?”

Function Point Discussion
Just for fun, let’s discuss Functional Measurement…

A General Concept
- Function Points
- Use Case Points
- Feature Points
- COSMIC Function Points

All attempt to measure “moving parts” of software
- Most of the calculations are difficult
- And require a design
- What can we learn to use with Story Points?

COSMIC Function Points
  - Relatively new, latest version is 2007
- Easy to Calculate
  - If we wanted to measure CFP production as we move along
  - Part of our Sprint Review “we produced 35 CFPs this Sprint”
- Easy to Estimate (very important for us)
- Only measures moving parts
  - Entry – sending information to the system
  - Exit – receiving information from the system
  - Read – reading information from persistent storage
  - Write – writing information to persistent storage
- See picture next page…
Counting COSMIC Function Points

- This is how FPs are counted using COSMIC
- Here is a typical system

For a given scenario, all we do is count the number of times information moves, and the total is our CFP.

But requests for data don’t count.

Yes, it is that simple...

Yes, it is that much of a pain...

Example: Add Traveler to Itinerary

- This is as simple as it gets
- A Small Story, with CFP = 2

Example:

- As a <traveler> I would like to add my daughter to my itinerary so that we could travel together.

Preconditions:
- Itinerary exists in CatAir DB
- Itinerary is up on screen

Postconditions:
- Itinerary with daughter added is in CatAir DB
- Itinerary on screen shows complete with me and my daughter
Example: Get List of Flights from CUTLASS

Get List of Flights
As a traveler I want a list of flights that match my itinerary so that I can find one that works for me
Preconditions:
- Traveler has prepared an itinerary
Postconditions:
- CUTLASS has returned a list of flights that matches the itinerary (up to 10)
- The list is presented to the traveler

- This is typical Medium-Sized Story, CFP = 4

Example: Pick One and Pay with VISA™

Pick Flight and Pay with VISA
As a traveler I want to pay with VISA for a flight that I pick from a list
Preconditions:
- a list of flights is on the screen
Postconditions:
- the chosen flight has been paid for
- the itinerary is updated as being paid for
- the payment is confirmed to the traveler

- This is a large one, CFP = 8
Summary of Story Size (Relative) Estimation

- Pick a “typical” M-Sized Story – possibly “a simple secondary scenario for an existing use case” with CFP = 4
- For Functional Stories, compare the story to the “basic M-Sized” Story
  - Use Estimation game with question: “How big is this one, in terms of moving parts, compared to our ‘typical M-Sized one’, given that the codebase is the same, the same people work on it, and so on”
  - Double (?) the points if it is “architecturally significant”
- For non-Functional stories with well-defined definitions of “done” compare the story to the “basic M-Sized” Story
  - Use Estimation game with question: “How hard is this one, in terms of effort, compared to our ‘typical M-Sized one’, given that …”
- For Stories with ill-defined definitions of “done”, timebox them
  - “Do 8 hours of Exploratory testing on page ABC”
  - “Do a Small Story’s worth of work cleaning up the code in module XYZ”

Our Solution
Baselines for Capabilities

- By working with Stakeholders and Developers, the team arrives at the following baseline sizes (in Function Point (FP)* for the wanted capabilities:

<table>
<thead>
<tr>
<th>Capability</th>
<th>FPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy an e-Ticket</td>
<td>150</td>
</tr>
<tr>
<td>Check Status of Flights</td>
<td>50</td>
</tr>
<tr>
<td>Get Hotel at Destination</td>
<td>100</td>
</tr>
<tr>
<td>Get Rental Car at Destination</td>
<td>100</td>
</tr>
<tr>
<td>Pilot Timesheets (risky)</td>
<td>250</td>
</tr>
<tr>
<td>Good Customer Plan (risky)</td>
<td>250</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>900</strong></td>
</tr>
</tbody>
</table>

* Function Points quantify the functional user requirements (FUR), independent of technical or quality requirements. FUR relate to (but are not limited to) data transfer, transformation, storage, and retrieval.

Counter-offer

- Standard estimation (20 hrs / FP) tells us that we'll need 18,000 Hours = 18 people for six months (there is no magic...)

- We go back to SirJeff and tell him that we would need a 18-person project team from day one, and we don't have that
  - We’d be glad to come up with a ramp-up plan for him
  - Or, we could figure out what we can actually give him with the people we have available, which is one team that is moving off of another project

- SirJeff says: "ok, tell me what this one team can give me in the next three months"
What’s Going On Now

- Have a team
  - Transitioning from SouvSite development
    - Half the team for the first 2 sprints
    - Whole team from then on
- SirJeff wants initial release in 3 months (7 sprints)
  - Needs to be useful to consumers
  - What he can tell his marketing and sales folks will be there?
  - This is a classic “Release Planning”-type question
- Remember, SirJeff’s priorities are:
  1. Buy an e-Ticket
  2. Check Status of Flights
  3. Pilot Timesheets
  4. The rest…

### Historical Information for Team

- 8 Members of Team
- Currently on SouvSite project
  - Velocity as below
- “Plannable” hours per 2-week sprint as at right

<table>
<thead>
<tr>
<th>Plannable Hours/Person</th>
<th>Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Hours</td>
<td>80</td>
</tr>
<tr>
<td>Minus Planning Day</td>
<td>72</td>
</tr>
<tr>
<td>Minus Grooming Mtg</td>
<td>68</td>
</tr>
<tr>
<td>Minus Lost Time</td>
<td>60</td>
</tr>
</tbody>
</table>

- Lost Time is vacation, sick, management time, etc
- Meetings aren’t plannable, as they are fixed time already spoken for
Baseline Total Capacity Calculations

- We have half the team for the first two sprints
- We have a Transition sprint, as we bring the rest of team over
- And then we have four “full” sprints
  - Joe’s on a Honeymoon sprints 5-6
- This is a total of 335 SPs as our baseline SP budget
- And we “spend” 2820 hours to do it
  - 60 hours/person/sprint

<table>
<thead>
<tr>
<th>Sprint</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours</td>
<td>240</td>
<td>240</td>
<td>480</td>
<td>480</td>
<td>468</td>
<td>432</td>
<td>480</td>
<td>2820</td>
</tr>
<tr>
<td>SPs</td>
<td>30</td>
<td>30</td>
<td>45</td>
<td>60</td>
<td>58</td>
<td>52</td>
<td>60</td>
<td>335</td>
</tr>
</tbody>
</table>

Adjusting the Count

- What we want, though, is how many SPs we can dedicate to SirJeff’s stuff… these will be our FPs
- So, we make adjustments to the count

<table>
<thead>
<tr>
<th>Step</th>
<th>What’s Going On</th>
<th>Avail SPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Calculate Total Capacity (what we just did)</td>
<td>335 SPs</td>
</tr>
<tr>
<td>2</td>
<td>Need a Release Sprint, so lose last Sprint’s worth</td>
<td>275 SPs</td>
</tr>
<tr>
<td>3</td>
<td>Lose SPs to Maintain SouvSite after Sprint 2 (10%, or 21 SPs)</td>
<td>254 SPs</td>
</tr>
<tr>
<td>4</td>
<td>Lose SPs for Chores (30%, or 76 SPs)</td>
<td>178 SPs</td>
</tr>
</tbody>
</table>

- So, we have 178 SPs to dedicate to SirJeff’s stuff in this release… (convert these to Function Points…)
- Let’s figure out how to use them
This is where it gets interesting

- Rely on the “wisdom of the crowds” again
  - Talk to team
  - Talk to Stakeholders
  - Innovation Games (Luke Hohmann)
- Do S/M/L estimation game at the Use Case level
  - Maybe budget 75/150/250 or 100/150/200 to the different sizes
- Maybe do some initial analysis and extrapolate using S-shaped curve
- Goal is to get a budget, and use agility to deliver within this budget
  - The low-level strategic agility the PO “owns”
  - Don’t want to cut yourselves short here – want to have a chance to succeed

S-Shaped Curve for Delivery of Use Cases...

- If we know SPs for “backbone”, can just multiply by 3
- If we know “all the stuff we want”, then just multiply by 3 (because we know the must-haves)
- Otherwise, make a guess based on size of Use Case (how many moving parts)
What the Team Thinks (more detail)

- **“Buy an e-Ticket”**
  - Backbone is 6 Large stories = 48 SPs, so need 144 SPs
  - Looks like an initial version of this is a M-Sized Use Case, so need 150 SPs
  - So, we want 150 SPs

- **“Check Status of Flights”**
  - Don’t really know how hard this is
  - If data is already in CUTLASS, is a really small Use Case
  - Willing to dedicate 50 SPs, but
    - 10 SPs dedicated to figuring out CUTLASS interface (need this anyway, for Buy an e-Ticket)
    - 40 SPs for Check Status, but no guarantee of making it

- **“Pilot Timesheets”**
  - Have no idea, but want to do 20 SPs worth of investigation to see how hard it is for next release…

Negotiating the Budgets (our baseline)

- Note that the Team thinks it needs 220 SPs, but the Capacity calculations show we only have 178 SPs to play with – Oops… Now what?

- Negotiation, arriving at the following table… and we also got Business Values from SirJeff for the ones he cares about…

<table>
<thead>
<tr>
<th>Capability/Item</th>
<th>BV</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy an e-Ticket</td>
<td>80%</td>
<td>108SP</td>
</tr>
<tr>
<td>Investigate CUTLASS interface/capabilities</td>
<td>10%</td>
<td>10SP</td>
</tr>
<tr>
<td>Investigate the basics of Pilot Timesheets</td>
<td>10%</td>
<td>20SP</td>
</tr>
<tr>
<td>Check Status of Flights</td>
<td>10%</td>
<td>40SP</td>
</tr>
<tr>
<td>SouvSite Maintenance (before Release Sprint)</td>
<td>10%</td>
<td>21SP</td>
</tr>
<tr>
<td>Chores (before Release Sprint)</td>
<td></td>
<td>76SP</td>
</tr>
<tr>
<td>Release Sprint (includes SouvSite Maint and Chores)</td>
<td></td>
<td>60SP</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>100%</td>
<td>335SP</td>
</tr>
</tbody>
</table>

**risky**

178
Finally, we have the info we need

- So, Release 1 Goal
  - Initial Release of CatAir website. Must be able to sell a ticket on the web
- Strategy:
  1. Learn How to use CUTLASS
  2. Get a minimal “Buy e-Ticket” as fast as possible
     - We know SP budget is risky
     - But this Epic is the most important
     - But don’t gold-plate – pay attention to “minimal”…
  3. Then, if Sprint 6 hasn’t begun yet, start “Status of Flights”
  4. Focus Sprint 6 on determining how hard “Pilot Timesheets” are
  5. If it’s possible to get a releasable version of “Status of Flights after investigating “Pilot Timesheets”, do so
  6. If not, get more functionality for “Buy e-Ticket” (make it prettier, for instance)
- Both the Goal and Strategy are negotiated, and agreed to, between SirJeff and the Scrum Team

And we Come Up With a Game Plan

- Note that we decomposed the Release Sprint into pieces, too, in order to manage “Release Activities"
Any Questions?

Monitoring the Release

Release BurnUp
Earned Value Metrics
Earned Business Value
Tracking SirJeff’s Release
How the Release Actually Played Out

How Release Played Out (SPs)

<table>
<thead>
<tr>
<th>Sprint</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours</td>
<td>253</td>
<td>233</td>
<td>495</td>
<td>530</td>
<td>467</td>
<td>445</td>
<td>498</td>
<td>2921</td>
</tr>
<tr>
<td>SPs</td>
<td>26</td>
<td>32</td>
<td>42</td>
<td>54</td>
<td>66</td>
<td>52</td>
<td>54</td>
<td>326</td>
</tr>
</tbody>
</table>

How Release Played Out (PersonHours)

Release Activities
- Buy an e-ticket
- Check Status of Flights
- Investigate Pilot Timesheets
- Investigate CUTLASS Interface
- SouvSite Maintenance
- Chores

Release Burnup

Correction In Sprint 3

- We are delivering capabilities and Stories
- But what we are managing is (largely) StoryPoints
- The BurnUp graph shows our production of StoryPoints
  - It shows our SP velocity graphically
  - It shows how many SPs we have “to go”
  - It shows our inventory of SPs that are “ready to go”
- And it’s easy to calculate
Earned Value Metrics (SPI and CPI)

The Metrics that Really Matter (CPI and SPI)

- CPI answers the question “are we paying what we expected for each SP?”
  - The ratio CPI = (baseline $/SP) / (actual $/SP)
  - In our case, $ is person-hours, and (baseline $/SP) is calculated cumulatively, sprint by sprint

- SPI answers the question “are we getting the SPs at the rate we expected?”
  - The ratio SPI = (actual SP/Sprint) / (baseline SP/Sprint)
  - In our case, our baseline velocity is also calculated cumulatively, sprint by sprint

- We like CPI and SPI to be ≥ 1 in standard EVM
- You’ll have to trust me on the derivations*

This graph shows our SPI and CPI as we move through the sprints. The values are calculated cumulatively, not one sprint at a time.

<table>
<thead>
<tr>
<th>Sprint</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs (B)</td>
<td>240</td>
<td>240</td>
<td>480</td>
<td>480</td>
<td>468</td>
<td>480</td>
<td>2820</td>
<td></td>
</tr>
<tr>
<td>SPs (B)</td>
<td>30</td>
<td>30</td>
<td>45</td>
<td>60</td>
<td>58</td>
<td>52</td>
<td>60</td>
<td>335</td>
</tr>
<tr>
<td>Hrs (A)</td>
<td>253</td>
<td>233</td>
<td>495</td>
<td>530</td>
<td>467</td>
<td>445</td>
<td>498</td>
<td>2921</td>
</tr>
<tr>
<td>SPs (A)</td>
<td>26</td>
<td>32</td>
<td>42</td>
<td>54</td>
<td>66</td>
<td>52</td>
<td>54</td>
<td>326</td>
</tr>
</tbody>
</table>

However...

- In an agile project, we are not paid to produce StoryPoints, we are paid to produce Business Value.
- Business Value is subjective, and based on our Stakeholder’s needs for this Release.
- Here are the features with BV in this release:

<table>
<thead>
<tr>
<th>Goal/Feature/Capability</th>
<th>BV</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy an e-ticket</td>
<td>80%</td>
<td>108 SPs</td>
</tr>
<tr>
<td>Investigate the basics of Pilot Timesheets</td>
<td>10%</td>
<td>20 SPs</td>
</tr>
<tr>
<td>Check Status of Flights</td>
<td>10%</td>
<td>40 SPs</td>
</tr>
</tbody>
</table>

As we deliver Stories within a feature, the Earned Business Value (EBV(feature)) increases, and is a percentage of the Feature’s Business Value (BV(feature)).
EBV Not Really a “Metric”

- It doesn’t measure anything
- It tells us where a capability’s EBV “should be” if
  - The overall budget for the capability was right
  - The PO is making good decisions about what is important
- In other words, an EBV graph is only good to start a conversation
  - According to this graph, this capability should be 65% “done”… is that what it feels like?
- So, the conversation about the EBV graph gives us very good information about progress of the project

For New Features

- That have an architectural element to them
- In our Case, we use this curve for
  - Buy an e-ticket
  - Check Status of Flights
For New/Existing Features

- Without an architectural element to add this Release
- In our Case, we use this curve for
  - Investigate the Basics of Pilot Timesheets

Earned Business Value Graphs

Correction
In Sprint 3
Any Questions?

Summary
Scrum Scales Up

- The “Wisdom of the Crowd” and the “Law of Large Numbers” allow us to some pretty sophisticated and accurate release Planning
  - But we must be agile as we do the work
  - It’s a baseline strategy, not a plan
- We can monitor against this baseline with some pretty good metrics

Any Final Questions?
Thank You Very Much!