Agenda

• Two Key Estimation Maxims
• “Typical” Agile Estimation
• A General Hypothesis…And Some Challenges
• A Supplemental (Alternative?) Approach
TWO KEY ESTIMATION MAXIMS
Two maxims are critical to effective estimation on software development projects

Any approach to estimation must facilitate communication when changes to the project can possibly impact cost and schedule of delivery

- Need to be able to explain the consequences to cost and schedule when requirements and/or development conditions change
- Communication, expectation management, and customer buy-in are critical to project success

Size matters!

- Software should be estimated based on some sort of sizing methodology and metric
- Estimating in level of effort (hours) does not enable effective communication
“TYPICAL” AGILE ESTIMATION
“Typical” Agile estimation is probably a unicorn
• Agile estimation methods vary widely
• Adopted and adapted by Agile teams

• Two concepts are consistently key to effective Agile estimation
  ▪ Collaboration
  ▪ Iteration
Let’s focus on story points
User Story Backlog

• User stories are short descriptions of a desired function or feature written from an end-user perspective
  ▪ Often expressed as, “As a user of this system, I want X feature so that I can accomplish Y”
  ▪ Serves as the basis (requirements) for what software functionality will be built
• Story points help teams assess the relative difficulty of the work they need to accomplish
  ▪ Typically starts with a “reference” story that serves as the basis of comparison to all other stories
  ▪ No “standards” for story points and they are determined on a team-by-team basis
Collaboration and iteration are essential to this Agile estimation approach.
• Planning poker provides opportunity for everyone on the product team to have input to size estimates based on their roles, perspectives, and experiences
  ▪ Teams often apply Fibonacci or other numerical sequences
  ▪ Discussion and iteration helps the team consider potential risk areas and develop a collective agreement about what stories to include in a sprint

• Estimating story points and applying velocity metrics can reduce biases and natural tendencies that typically occur when estimating level of effort in hours
A GENERAL HYPOTHESIS...AND SOME CHALLENGES
Hypothesis: In general, story point estimation works well for Agile teams for planning sprints.
IMMEDIATE FEEDBACK
The immediacy of feedback of data from the previous sprint during retrospectives provides the opportunity for lessons learned to be applied right away.

CALIBRATION
The relative nature of story points allows teams to tailor and calibrate the size unit to their own situation.

COLLABORATION and ITERATION
Collaboration and iteration with all stakeholders, including the customer/product owner encourages communication and expectation management.
Challenges

**BUDGETING**
Generation of estimates to establish initial project budgets

**NEW SCRUM TEAM**
Formation of a new development team with no history together

**IT PORTFOLIO MANAGEMENT**
Establishment of organizational portfolio management with consistent metrics across projects

**ASSUMING VELOCITY = PRODUCTIVITY**
Team comparisons, improvement goals, etc. Be prepared for inflated story point estimates and other fiction

**BENCHMARKING**
Cannot perform any kind of reasonable benchmarking analysis either within the organization or against industry data

**ESTIMATING IN HOURS**
Using points as a proxy for hours is no different than estimating in hours and eliminates the benefits of using relative sizing. Example: 1 story point equates to 4 hours of work

**DIFFERING STORY POINT DEFINITIONS**
Customers may have an alternate idea of what a story point is or should be. Without an established “standard” for story point, this disconnect can be a real possibility. Can be a source of serious miscommunication and misunderstanding
A SUPPLEMENTAL (ALTERNATIVE?) APPROACH
Let’s talk about an estimation approach designed to help overcome story point challenges
This methodology can either supplement or replace Agile story point estimation for sprint planning

Supplement
- If your team needs a more effective way to develop defendable initial project estimates or a set of consistent metrics

Replace
- If you find that your team has difficulty applying story point estimation consistently or is subject to some of the pitfalls that can happen

So…if only there was an industry standard software sizing measure that is based on functionality described from the users’ perspective…
The function point standard established and maintained by the International Function Point Users Group is exactly that:

- Measures software size based on the functional requirements requested by and provided to the user.
- Accepted as a standard size measure by ISO (20926:2009).
Basic methodology: identify data and transaction functions, assess the complexity, and apply the FP matrix

- Data function complexity determined by the number of data elements and logical data subgrouping
- Transaction function complexity determined by the number of data elements and files referenced
- The IFPUG FP matrix identifies how many function points each identified function receives

<table>
<thead>
<tr>
<th></th>
<th>Low</th>
<th>Average</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Logical File</td>
<td>7</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>External Interface File</td>
<td>5</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>External Input</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>External Output</td>
<td>4</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>External Inquiry</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>
The real problem with function points: perception
The term “function points” can evoke a visceral negative reaction from many people, especially Agile enthusiasts.

- **Perception:** Function points take too much time and effort

- **Perception:** Requirements must be fully defined to effectively apply function points

- **Perception:** Function points don’t offer the flexibility we need to estimate in an Agile environment

- **Perception:** Function points are not granular enough to apply in an Agile environment
Proposal: use an alternative (but similar) sizing measure for Agile projects: **Agilons**
COMPLEXITY
Low, Average, High complexity generally determined by how complex the team thinks the particular story will be; consider number of expected data elements, difficulty of function, etc.

USER STORY CROSS CHECK
Provides a good litmus test for user stories as well: if a user story can be applied to more than one type of Agilon, it probably should be broken down into simpler stories (concept of elementary process or minimum viable product)

VELOCITY
Velocity can be measured in Agilons per sprint
Standard Agilon Matrix

<table>
<thead>
<tr>
<th></th>
<th>Low</th>
<th>Average</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Data</td>
<td>7</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>External Data</td>
<td>5</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>Input</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Output</td>
<td>4</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Inquiry</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

Not too dissimilar from...a Fibonacci sequence?
As a customer I would like to have the ability to search for and reserve a hotel room in order to spend the night in another city.
For simplification, let’s assume “average” complexity for any identified Agilons

• User story seems to have multiple Agilon types that need to be decomposed
  ▪ Hotel data (Internal Data - 10 Agilons)
  ▪ Search for hotel room (Inquiry – 4 Agilons)
  ▪ Reserve hotel room (Input – 4 Agilons)

• Total of 18 Agilons

• If my team’s velocity is around 18 Agilons per sprint, we’re good to go…

• If not, perhaps we need to decompose the user story a bit
So how does this approach really address the challenges of story point estimation?
• **Initial budgets** – estimates can be fully documented and explained, even in the absence of requirements, and then facilitate communication

• **Formation of a new Agile team** – standardized sizing metric, combined with good historical data or a parametric model, can provide estimates that stakeholders can understand and be more comfortable with

• **Organizational metrics** – a standardized size measure empowers consistent productivity and quality metrics across an organization, offering real possibilities for improvement
Sources:


http://www.construx.com/uploadedFiles/Construx/Construx_Content/Resources/Presentation/AgileEstimation_KeyPrinciplesAndPracticesforSuccessfulAgileProjects.pdf


https://www.mountaingoatsoftware.com/blog/the-main-benefit-of-story-points

https://help.rallydev.com/top-10-mistakes-teams


https://www.sitepoint.com/make-7-mistakes-agile-estimation/

IFPUG, “Applying Function Point Analysis to Scrum Agile Software Development Projects,” V.1.0, October 25, 2013


https://www.industriallogic.com/blog/stop-using-story-points/

https://www.atlassian.com/agile/estimation