Thank you for having me today.
I am Rick Anderson from Wind River located here in Beaverton, Oregon
I am the Engineering Director of our Solutions Technology Center, which produces products primarily for Android.
I’ve been running Scrum teams at Wind River for four years now and before that for about four years at Tektronix.
The title of our talk is a bit catchy, but it’s really about using Scrum with teams that are not co-located.
So let’s begin.
Important that you read this slide because my presentation has nothing to do with BunnyPeople or skool, although that is a very cool word with three O’s in it
The Problem

- Scrum is an Agile software development + project management methodology
- Scrum was designed for co-located teams
- Unfortunately teams are often not co-located today
- Trending…
  - Companies continue to expand geographically either through acquisitions or chasing low cost resources
  - More employees are working from home to save office space costs, reduce pollution or for quality of life reasons (reduce commute time)
  - Large corporations are hiring the best talent regardless of location
- So how can you marry Scrum with dispersed teams?

So this is the problem we will be talking about today
Will not be covering any details Agile or Scrum today; it is assumed that you know at least a little about this. There is some material in the backup slides. But even if you are not Scrum literature, if you are working with remote teams, I think you will benefit from listening to this presentation because many of the lessons apply to anyone working with a remote team regardless of methodology.
So a little bit more about me...

Your Speaker

• I've been managing remote teams since 2000, a practicing ScrumMaster since 2006, certified since 2008

• I've been told you cannot run Scrum across the ocean

• I would love for my team to be co-located
  – But I don’t expect this anytime soon

• I’m different
  – I’ve run Scrum teams...
    – located at ten different sites...
    – on three different continents...
    – in seven different times zones
  – I’ve run ten week sprints
  – I’ve run Scrum teams with over 25 people on them
"Dispersed" Teams

- Not co-located (one or more workers are remote, or not face-to-face)
  - Multiple sites in same country
  - Multiple sites in different countries
  - At home workers (Small Office Home Office)
  - Traveling workers (maybe your Product Owner)

- We will assume time zone and geography challenged
  - Consider a near worse case scenario: workers in Romania (5pm), East Coast (10am), West Coast (7am), China (10pm) and Korea (11pm)

Let’s define what a dispersed team is
How many are practicing Scrum across dispersed teams today?
Most teams that are doing Scrum and are no co-located are following one of these three methods
Each has advantages and disadvantages

Scrum with Dispersed Teams

• Typical Scrum solutions with dispersed teams:
  – **Integrated Scrum** – A single sprint with dispersed members participating just like they were co-located
    – If geo dispersed, rarely practical due to time zone differences
  – **Isolated Scrums** – No commonality, coordination; each site runs their own isolated sprint
    – Has many disadvantages
    – Rarely practical because tasks not easily separated; dependencies
  – **Scrum of Scrums** – Each team runs it’s own sprint, but they are synced; outputs roll-up to master sprint backlog
    – Can be complicated
    – Potentially not as synergistic
So let’s talk about a fourth method, and the one I’m practicing.

**Scrum with Dispersed Teams**

- Hybrid of Integrated Scrum and Scrum of Scrums seems best
  - Let’s call it a **Dispersed Scrum**!
  - A single sprint across all sites (like an Integrated Scrum)
  - But run locally by each local team (like Scrum of Scrums)
    - One ScrumMaster per site
    - They remove roadblocks for that site or can be solved by that site
  - Local Daily Stand-Up meeting (single Sprint Backlog)
  - Planning takes place over 24-48 hours (each site contributes)
    - User stories assigned to specific individuals
    - More on planning in a second...
  - Demo and Retrospective meeting - attempt to get team together
  - Not complicated, synergistic, and over comes time zone issues
If you didn’t follow the last two slides, here is another way of looking at it...

<table>
<thead>
<tr>
<th><strong>Method</strong></th>
<th><strong># of Sprints</strong></th>
<th><strong>Scrum-Master</strong></th>
<th><strong>Integrated</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrated Scrum</td>
<td>1 😊</td>
<td>At main site only</td>
<td>Yes 😊</td>
</tr>
<tr>
<td>Isolated Scrum</td>
<td>2 or more</td>
<td>At local site</td>
<td>No</td>
</tr>
<tr>
<td>Scrum of Scrums</td>
<td>2 or more</td>
<td>At local site + head SM site</td>
<td>Yes 😊</td>
</tr>
<tr>
<td>Dispersed Scrum</td>
<td>1 😊</td>
<td>At local site</td>
<td>Yes 😊</td>
</tr>
</tbody>
</table>
Dispersed Scrum: Shortcomings

- While a Dispersed Scrum is better, it is not perfect
- It's not face-to-face
- Planning, demo, retrospective meetings can still be time zone challenged
- Planning takes place over a 24-48 hour period instead of in four hours
- Interactive/Delphi estimation (Planning poker) is replaced by less effective peer review

Still not perfect.
But more appealing.
Planning a Dispersed Scrum takes a little more planning, time and coordination.
So let’s turn to some of the keys to success for doing Scrum with dispersed teams. These are not necessarily in priority order!

**Keys to Success (tips)**

1. Crisp team communication (3 / 2)
2. Accountability (5)
3. Daily Stand-Up meeting (3)
4. Strong local site management (1)
5. Self organizing and empowered teams (2)
6. ScrumMaster issues (5)
7. Product Owner issues (3)
8. Track plan versus actual (1)
9. SQE team (6)

*Backup Slides have the blue Keys to Success*
Crisp Team Communication

• **Challenge #1:** Scrum and agility usually require a faster pace
  – This fast pace and stress some team members
  – Often exposes communication weaknesses in a group

• **Challenge #2:** Scrum requires daily interaction
  – Yet time zones and distance get in the way
  – Doesn't help that 90% of what we communicate is non-verbal
Crisp Team Communication

- **Techniques for Success (TforS) #1**: All email must be acknowledged within your working day
  - Ask people to check email early in morning and late at night to increase turnover rate

- **TforS #2**: Over communicate
  - Executive summary + details + quantitative data + links
  - Document everything in a public place
  - Work for perfection – “perfect communications every time”

- **TforS #3**: Standardize on communication tools, status report templates, etc.
  - Beyond “work done yesterday, work today, roadblocks”, what does SM and PO need to know? Be very clear about what is required.

Use of Twiki’s, daily email status reports, spreadsheets with quantitative data
Accountability

- **Challenge #1:** Scrum exposes best/worse estimators
  - Off-shore resources often don’t have same estimation experience (younger, not as much education in estimation)

- **Challenge #2:** Scrum exposes high/low productivity
  - It is more difficult to hide *(not specific to dispersed team, but can bring into question value of off-shore resources)*
Accountability

- **TforS #1**: Discuss what accountability and schedule ownership means
  - Solid bottom-up estimation up front, including review by others
  - Make it acceptable to add tasks, raise roadblocks
  - Work with those who are not doing well

- **TforS #2**: Emphasize schedule every day (burndown chart)

- **TforS #3**: Don’t plan for 100% of your available hours during planning meeting
Accountability

- TforS #4: Use six hour work days (two overhead hours)

- TforS #5: Four hour rule
  - If you've spent four hours trying to solve the problem and you are unsuccessful, ask for help (raise the issue as a roadblock)
Daily Stand-Up Meeting

• Challenge #1: A co-located daily stand-up meeting is impossible with a dispersed team
Daily Stand-Up Meeting

• **TforS #1:** Use the Dispersed Scrum model
  – Have multiple daily stand-ups with a ScrumMaster in each region or at each major site
  – Use a single sprint backlog, burndown, roadblocks list
  – Store in common accessible place
  – Make sure it's under CM control and backed-up
  – Everyone still sees everyone else's tasks and progress
  – Must be religious about updating and communicating
Daily Stand-Up Meeting

- **TforS #2**: Use the medium that works best for your team given locations, bandwidth, etc.
  - Video (video conference, Skype, WebEx)
  - Recorded video
    - Store in common accessible place
  - Text Messaging (IRC, IM, Chatrooms)
    - Have status typed-up ahead of meeting start
    - Use an agreed upon format
      - Example: task / hrs yesterday / hrs remaining (162/4/13)
    - Can overcome some language issues

- **TforS #3**: Continue to have a weekly group meetings
  - Different focus, pace and interaction than Daily Stand-Ups
ScrumMaster Issues

- **Challenge #1:** Hard for ScrumMaster (or Product Owner) to manage and interact with a dispersed team
  - Might have to attend daily stand-ups during non-working hours
  - Hard to get a real feel for status when you can’t see and touch it (demo)
  - Don’t hear the hallway conversations or see the panic on their faces

- **TforS #1:** Have one SM per site (Dispersed Scrum model)

- **TforS #2:** Use the right communication mechanism for the job (often phone -> Skype -> IRC -> email - in that order)

- **TforS #3:** Work in remote site time zone at times and visit the site at least once a year
ScrumMaster Issues

- **TforS #4**: Rotate ScrumMaster between those that have the right traits
  - This strengthens your team
  - No reason why non-SWEs can’t be the SM (SQE, TechPubs, ...)

- A good SM must be:
  - Organized and prompt
  - Helpful
  - Patient and flexible
  - Accountable
  - Be detail oriented
  - Have good communication skills

- **TforS #5**: Learn the local language, customs, culture
Product Owner Issues

- **Challenge #1**: The Product Owner is watching closely
  - Can be intimidating to those not used to working with a real customer or a strong marketing person (*not specific to dispersed teams*)

- **TforS #1**: Make sure your Product Owner understands their role and the influence they have
  - Level on time commitment, participation expectations, etc.

- **TforS #2**: Supplement primary PO with a local engineering team manager or architect / technical leader
Product Owner Issues

- TforS #3: Make adjustments to the product demo meeting so it is still relevant for all
  - Have demos at all sites
  - Use video
  - Celebrate milestones as a team
Track Plan versus Actual

• **Challenge #1:** Scrum normally tracks the original estimate and hours remaining
  – This makes it hard to identify slipping tasks and those who have estimation problems (**not specific to dispersed teams, but if you are remote, you want to easily tell what changed every day**).
# Track Plan versus Actual

- **TforS #1**: Track actual work hours  
  - It’s easy to do!

<table>
<thead>
<tr>
<th>Backlog Item</th>
<th>Task</th>
<th>Resource</th>
<th>Task Status</th>
<th>Initial Est’d Hrs</th>
<th>Hrs Worked</th>
<th>10-Sep</th>
<th>11-Sep</th>
<th>12-Sep</th>
<th>13-Sep</th>
<th>14-Sep</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jelly Bean Security</td>
<td>Test and debug</td>
<td>Rick</td>
<td>Not Started</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Add alpha blending feature</td>
<td>Rick</td>
<td>Completed</td>
<td>12</td>
<td>18</td>
<td>12</td>
<td>12</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Gingerbread security exploit</td>
<td>Rick</td>
<td>In Progress</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Combine existing code &amp; open source</td>
<td>Rick</td>
<td>Completed</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Video</td>
<td>Rick</td>
<td>Not Started</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Demo image creation</td>
<td>Rick</td>
<td>Other</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>57</td>
<td>18</td>
<td>57</td>
<td>51</td>
<td>45</td>
<td>45</td>
<td>25</td>
</tr>
</tbody>
</table>
Final Thoughts

- Building a high performing dispersed Scrum team takes...
  - Lots of work (patience)
  - Calendar time (it won't happen overnight)
  - Trust (empower the team, hold them accountable)

- Consider using the Dispersed Scrum model

- Considering adopting some of the Techniques for Success presented in this presentation

- Create an environment where “We are all in this together”

- Use retrospectives as constructive events

- Celebrate success!
Questions / Answers
Backup
Scrum Overview

- Scrum is an Agile Software Development *and* Project Management methodology

- WikiPedia has a nice history:
  - 1986 by Hirotaka Takeuchi and Ikujiro Nonaka
  - 1993 by Dr. Jeff Sutherland and 1995 by Ken Schwaber
  - 2001 Agile Manifesto
  - 2001 "Agile Software Development with Scrum" by Schwaber and Beedle

- Roles:
  - Product Owner
  - Scrum Master
  - Development Team
Scrum Overview

• Concepts:
  – Sprint Planning Meeting
  – Time-boxed
  – Velocity and Story Points
  – Daily Stand-up Meeting
  – Sprint Review (Product Demonstration, Retrospective)

• Process and Product Assets:
  – Product Backlog
  – Sprint Backlog
  – Burndown Chart
  – A Working Product!
Scrum Overview

Crisp Team Communication

- **Challenge #3:** Scrum means everyone must be involved
  - More difficult for the introverted engineer

- **TforS #4:** A team of equals
  - Everyone must speak up (no Jr. Engr -> Sr. Engr)
  - No filtering of data (Sr. Engr -> Jr. Engr)

- **TforS #5:** Cultural training (if necessary)
  - A close working team understands each others cultures
  - Provide financial support for training
Self Organizing Teams

- **Challenge #1:** Problem solving must be pushed down to the individual engineer with dispersed team
  - With a co-located team, there is a knowledge pool sitting around you and it is easy for a strong Engineering Manager, Technical Leader or ScrumMaster to solve team problems
  - The number of common working hours might be limited
Self Organizing Teams

• **TforS #1:** To really build a strong team, you must let the team figure out some things themselves - teach problem solving techniques
  – Fishbone diagrams
  – 5 Why’s
  – Share lessons learned

• **TforS #2:** When there is a problem, schedule a meeting and act as a moderator, while letting the team figure it out
  – Don’t solve the problem, but guide problem solving
**Strong Local Site Management**

- **Challenge #1:** Can be hard for leaders to understand what is going on at the remote site
  - No eyes
  - No ears
  - No one person you can go to for stuff that falls inbetween typical job duties
  - If international, understanding of local laws and culture
Strong Local Site Management

- **TforS #1**: Having a strong designated leader at your remote site helps greatly
  - They can act as the local translator
  - They can be your eyes and ears
  - Find and keep that leader!
  - Recognize their importance to your success
  - Have weekly 1:1’s
SQE Team

• **Challenge #1:** Software Quality Engineering (SQE) is often off-shored and therefore dispersed

• **Challenge #2:** Is SQE on the Scrum team or not?
  – The answer to this seems obvious (*not specific to dispersed teams*)
SQE Team

- TforS #1: Have an SQE function
  - This remains a best practice
- TforS #2: Have SQE be part of the Scrum team
  - Separate group ok, but 100% integrated
- TforS #3: They should be equals with the Software Engineers
- TforS #4: Identify Software Quality Leaders
- TforS #5: Push for test automation
- TforS #6: Part of their role should to be push quality upstream (software process improvement)

Moderate retrospectives, take actions to improve quality, etc.
Lessor Keys to Success

- Use Appropriate Tools
  - Rally (in the cloud...)
  - Excel or Open Office Spreadsheets
  - Facilitate Pro (great for dispersed team Retrospectives...)

- Daily Builds
  - Automated, email notification, buildmeister, used daily

- Good Configuration Management techniques
  - Gatekeeper role, code reviews, tools, etc.

- Visit your dispersed team twice a year
  - Or have them come visit you

Will not be talking about these further in this presentation