Game Play

• Why Play Games
• What are Agile Games
• Talk about some
• Play two
Learning Pyramid

Hear one, see one, do one, teach one
Agile Games

Interactive learning sessions that amplify some learning objective.

Group activity to accomplish some specific objective.
Ball Point Game

1. Allow the team two minutes of preparation time to determine how they will organize themselves.
2. Get an estimate from the team of how many balls they can pass through the system.
3. Run a two-minute iteration.
4. Allow the team one minute to discuss how to improve the process.
5. Repeat for a few (five) iterations.

http://scrumology.com/from-the-archives-the-ball-point-game/
Ball Point Game

• Insights
  Demonstrate self organization
  Natural velocity
  Power of inspect and adapt for process improvements.

http://scrumology.com/from-the-archives-the-ball-point-game/
Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck    James Grenning    Robert C. Martin
Mike Beedle   Jim Highsmith    Steve Mellor
Arie van Bennekum  Andrew Hunt    Ken Schwaber
Alistair Cockburn  Ron Jeffries    Jeff Sutherland
Ward Cunningham    Jon Kern    Dave Thomas
Martin Fowler    Brian Marick

Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information about and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity—the art of maximizing the amount of work not done—is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
Pocket-sized Principles

1. Divide participants into groups.

2. Challenge each team to, within a 15 minute time-box, come up with **three words maximum** that effectively capture each of the twelve principles.

   For Managers: The actions I will take to make this happen.

3. When time is up, go through each principle and discuss which are the most important words.

http://tastycupcakes.org/2010/01/pocket-sized-principles/
Pocket-sized Principles

1. Produce Value Early
2. Welcome Change
3. Iterative Delivery
4. Daily Business Collaboration
5. Trust Motivated Team
6. Face to Face
7. Working Software
8. Sustainable Pace
9. Technical Excellence
10. K.I.S.S.
11. Self-Organize
12. Reflect and Adjust

http://tastycupcakes.org/2010/01/pocket-sized-principles/
Marshmallow Challenge

• Teams must build the tallest free-standing structure out of 20 sticks of spaghetti, one yard of tape, one yard of string, and one marshmallow.

• The marshmallow needs to be on top.

http://marshmallowchallenge.com/Welcome.html
Marshmallow Challenge

• Team building: builds rapport.
• Ask for insights:
  • Who tends to do the worst? Why?
  • Who tends to do the best? Why?
  • What improves performance?
  • What kills it?
• Scientific method used in innovation.
  • Goal, hypothesis, test, measure, repeat.

http://marshmallowchallenge.com/Welcome.html
Kanban Pizza Game

• When introducing Kanban it may be difficult to convey concepts of WIP and flow.

• Objective: create paper pizzas as fast as possible. First time, no structure. Then, with a flow system. Later, with WIP limits.

• Insights: slow down to increase throughput.

http://www.agile42.com/en/training/kanban-pizza-game/
Buy a Feature

• Create a list of potential features and provide each with a price.
• Make certain that some features are priced high enough that no one customer can buy them.
• Give customers “monopoly money” that is 1/3 to ½ of what’s necessary to buy all features.
• Customers buy features that they want in the next release.
• Encourage customers to pool their money to buy especially important and/or expensive features.
• Record why the selections were made.

http://www.innovationgames.com/buy-a-feature/
San Jose Budget Games

• Lists of potential cuts (with savings) and additions (with costs).

• Tables of 8-10 with 1 facilitator review proposals.
  • To fund a proposal, unanimous consent is NOT required.
  • To reduce a proposal, unanimous consent IS required.

• Each table prioritizes the proposals.
  Tax increases are considered.
Budget Games Results

- Residents routinely explain choices in terms of “systems thinking”
  - The influence of roads on public safety
  - The impact of park rangers on police
- Residents considered both near-term and long-term impacts when discussing choices
- During the second year, 80% of recommendations adopted into city budget.
Forging a path to Paradise

Why Retrospectives fail and what you can do about it

Jay Packlick
Destination: Your Island

1. Steps **YOU** personally can take to get closer to the island

2. Steps your **TEAM** can take to get closer to the island

3. Steps your team can **INFLUENCE** to get closer to the island

You are here (So far)

The water....
Herculean Doughnut

- Confusion of rolls and responsibilities?
- Get everyone in the team to arrange the cards on the board. Ask “Where do you think this card goes?”
- If a responsibility is for one section of the team, it is placed in its specific area.
- If two or more share it, it goes on the dividing line between them.

http://tastycupcakes.org/2013/02/the-herculean-doughnut/
<table>
<thead>
<tr>
<th>Action</th>
<th>Who</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1. Articulates the Product Vision</td>
<td>Product Owner</td>
<td></td>
</tr>
<tr>
<td>A2. Represents the Business and Customers</td>
<td>Product Owner</td>
<td></td>
</tr>
<tr>
<td>A3. Writes User Stories</td>
<td>Entire Team</td>
<td>In fact, anyone can write stories and add them to the Product Backlog. The Product Owner will order the items in the Release and Product Backlog.</td>
</tr>
<tr>
<td>A4. Recognizes Impediments</td>
<td>Entire Team</td>
<td></td>
</tr>
<tr>
<td>A5. Manages the Team</td>
<td>No One</td>
<td></td>
</tr>
<tr>
<td>A6. Accepts a Story as Meeting the Acceptance Criteria</td>
<td>Product Owner</td>
<td></td>
</tr>
<tr>
<td>A7. Protects the Team from Outside Distractions</td>
<td>Scrum Master</td>
<td></td>
</tr>
<tr>
<td>A8. Commits to complete the Sprint Backlog by the end of the Sprint</td>
<td>No One</td>
<td>The word commit comes with too much baggage and can lead to teams completing work with lower quality. The Scrum Guide changed from the team commits to the team forecasts. Another way to think of it... The team agrees and believes it can complete the Sprint Backlog by the end of the Sprint.</td>
</tr>
<tr>
<td>A9. Points out Scrum Team Member’s Mistakes</td>
<td>Entire Team</td>
<td>Focus is on behavior, not attitude or personalities.</td>
</tr>
<tr>
<td>A10. Creates Project Budget</td>
<td>Project Manager</td>
<td></td>
</tr>
<tr>
<td>A11. Motivates the Team</td>
<td>Product Owner</td>
<td>The greatest motivation comes from elevating goals. Things like “Fred from Fidelity wants to spend weekends with his family instead of &lt;something&gt;. We need to solve this problem for Fred.” The Product Owner is the one who brings this message to the team.</td>
</tr>
<tr>
<td>A12. Remove Impediments</td>
<td>Scrum Master</td>
<td>Could get help from anyone in the organization.</td>
</tr>
<tr>
<td>A13. Inspects and Adapts to Improve their Performance</td>
<td>Entire Team</td>
<td>Usually done in the Sprint Retrospective.</td>
</tr>
<tr>
<td>A14. Creates the Release Plan</td>
<td>PO and Dev Team</td>
<td>The Release Plan is created through collaboration between the PO and Dev Team. The PO owns the priorities of the items while the Dev Team sizes the stories and estimates their velocity.</td>
</tr>
<tr>
<td>A15. Cancels a Sprint</td>
<td>Product Owner</td>
<td>Usually after consulting with the dev team.</td>
</tr>
<tr>
<td>A16. Sizes the Stories</td>
<td>Dev Team</td>
<td></td>
</tr>
<tr>
<td>A17. Facilitates Meetings</td>
<td>Scrum Master</td>
<td></td>
</tr>
<tr>
<td>A18. Orders the Release Backlog</td>
<td>Product Owner</td>
<td></td>
</tr>
<tr>
<td>A19. Rejects a Story as not Ready for a Sprint</td>
<td>Dev Team</td>
<td></td>
</tr>
<tr>
<td>A20. Builds the Initial Product Backlog</td>
<td>Product Owner</td>
<td></td>
</tr>
<tr>
<td>A21. Chooses the Amount of Work in a Sprint</td>
<td>Dev Team</td>
<td></td>
</tr>
<tr>
<td>A22. Keeps Stakeholders Informed</td>
<td>Product Owner</td>
<td></td>
</tr>
<tr>
<td>A23. Ensures Something Useful is Built by the Launch Date</td>
<td>Product Owner</td>
<td></td>
</tr>
<tr>
<td>A24. Assigns tasks to Team Members</td>
<td>No One</td>
<td>Dev Team members claim tasks and stories for themselves.</td>
</tr>
</tbody>
</table>
Herculean Doughnut, Larger Organization
What are your favorite Agile Games?

Responses were...

• Helium Stick
• Penny Game
• Napkin (given a cloth napkin, make something)
• Paper folding Kanban system.

• ScrumLiner (paper airplane)
• Jigsaw puzzles with pieces in different boxes.
• Product Box
• Faces and Facts
• 2 truths and a lie
Sizing Stories
Sizing User Stories with Story Points

• The “bigness” of the story:
  • How hard it is.
  • How much of it there is.
  • Includes ALL the work to complete the story, including documentation, QA, etc.

• Story points are unit-less. They don’t measure effort, duration, only size.

• Story points are relative estimation, comparing the size of one story to another:
  • If a login screen is a 2, then the search feature is an 8.
  • Studies have shown we are better at relative estimation than absolute estimation

SOURCE: An Introduction to Agile Estimating and Planning, Mike Cohn, 12 March 2008, mountaingoatsoftware.com
Exercise: Estimate the Size of Each

In Pixels
Exercise: Estimate the Size of Each

Relative to One Another

If this box is a 1, how big is the box on the right?
Estimate Only Within One Order of Magnitude

If the box above is a 1, how big is the box on the right?
Team Estimation Game

- Place post-its on table labeled 1, 2, 3, 5, 8, 13, 20, 40; may use 100, 500, 1000 if large stories (epics) exist.

- Someone starts by picking up a story, reads it aloud, and places under their estimated size of story (based on level of effort, not time). 1 is easy; 13 is difficult; 20, 40, ... are too big – story will need to be broken down.

- Second person reads next card aloud; compares it to the first card; and places their card under the number of their estimate (relative to the first).

- Third person and every one after that... CHOOSE or MOVE... choose to pick up, read aloud and place a new story down OR move an existing card to a new value.

- Continue until all cards are down and team comes to consensus (which means nobody is in violent opposition to moving forward).
Insights

• Did anyone ask for clarification on the cards?

• Did both teams get the same results? What does that say about comparing teams and their velocities?
<table>
<thead>
<tr>
<th>One Point</th>
<th>Two Points</th>
<th>Three Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Five points</td>
<td>Eight Points</td>
<td>Thirteen Points</td>
</tr>
</tbody>
</table>
Planning Poker

• A quick, iterative way to do estimation.
• Full team participation.
• Allows teams to come to a consensus quickly.
• Can be used during Backlog Refinement to size new items relative to prior backlog.

SOURCE: An Introduction to Agile Estimating and Planning, Mike Cohn, 12 March 2008, mountaingoatsoftware.com
Multi-tasking
Exercise: Need 4 to 5 Volunteers

A, B, C, ... Z

3, 6, 9, ... 45

5, 10, 15, ... 60
Insights

What happened? How did that feel?

What happened to the quality?

Credit: Peter Saddington MDiv, CST, Action & Influence, Inc
Debrief

What happened? (let them respond) What we’re looking for...

As the team multi-tasked, time went up and quality went down.

And this is something simple. We’re just reciting a series of letters and numbers. Imagine what happens when you multitask on something complicated like writing code!

How did that feel? (let them respond) You may get “confused, under pressure”. I’ve even gotten “stupid”.

Reassure them that this is a natural response.

Multitasking often results in lower job satisfaction.
Did you know?

• A 2001 study reported in the Journal Of Experimental Psychology found:
  • It took students up to 40% longer to solve complicated math problems when they had to switch to other tasks.
  • Doing so caused the release of stress hormones and adrenaline.

• A 2008 study by Gloria Mark (U of California, Irvine) showed:
  • When people are frequently diverted from one task to another, they work faster, but produce less.
  • They reported significantly higher stress levels, frustration, workload, effort and pressure.
Increase Productivity by Reducing Multitasking

- Sequential work yields results sooner when multitasking is eliminated.

- Not multi-tasking but task-switching, and it takes more time to switch tasks than stick with them until you finish.
Stop Starting and Start Finishing

Source: NetObjectives
Jeff Sutherland @ Scrum, Inc: A Pattern Language for Hyperproductivity

Jeff Sutherland, the inventor of Scrum, is a charter member of the Scrum Patterns group. He is the author of most of these Scrum PLoP® patterns — patterns he teaches to get teams off to a good start, and to get great teams to a hyperproductive state:

- How do you get started? (Stable Teams)
- How do you successfully pull backlog items into a Sprint? (Yesterday's Weather)
- How do you get stuff done? (Swarming: One-Piece Continuous Flow)
- How do you deal with interruptions during the Sprint? (Illegitimus non Interruptus)
- How do you get defect free at the end of the Sprint? (Daily Clean Code)
- How do you deal with surprises? (Emergency Procedure)
- How do you ensure you continuously improve? (Scrumming the Scrum)
- How do you get teams to have fun? (Happiness Metric)
- How do you get hyperproductive? (Teams that finish early accelerate faster)

http://www.scrumplop.org/
Where to Find More Agile Games?
More from David Koontz

Agile Complexification Inverter

Thursday, March 27, 2014

List of Agile Team Exercises

A collection of blog post that describe team exercises - if you need more information about one of these please contact me.

Video series on Scrum (short takes)

http://agilecomplexificationinverter.blogspot.com/2013/10/learn-scrum-video-series.html

About

David's thoughts, rants, and ramblings on Agile software development theory and practice.

David Koontz is a practicing Agile Transition Guide. He practices because it is the only way to get to Carnegie Hall.

http://agilecomplexificationinverter.blogspot.com/2014/03/list-of-agile-team-exercises.html
Monday, October 19
Capital One Campus
Early Bird: $295 until TOMORROW @ NOON
William “Red” Davidson

PMP  PMI-ACP  CSM
CSPO  ICP-ACC  SAFe-SA

reddavidson@yahoo.com