Kanban Kickstart Geeknight

Jesper Boeg, Agile/Lean Coach,
VP Trifork Agile Excellence
jbo@trifork.com
Twitter: J_Boeg
Agenda

- Introduction + Advertisement (15 min.)
- A Couple of War Stories
- The 5 Principles of Kanban
- Real Life Experiences
- Break: Sandwich and Networking
- More Real Life Experiences
- Discussion Topics
What I Won’t Cover

- Kanban’s Origins
- Kanban principles In-depth
- A Detailed Comparison of Scrum vs. Kanban
GOTO; Cph Conference

- Cool Products, Technical Tracks, Architecture
- Agile
  - Mærsk Case Study, Agile Games, David Snowden
- Lean
  - Benjamin Mitchell, Don Reinertsen, (Jesper Boeg)
- Agile UX
  - Chris Nodder, Janne Jul Jensen (Winner of 2 Danish App Awards)
- DevOps/Continuous Delivery
  - Patrick Debois
Agile Coaching and Training
- Scrum, Kanban, Lean, XP....

Software development:
- Public, Mobile, Security and Finance

Technical Training
- Mobile, Java, .Net....

Conferences
- GOTO and QCon
Popular Agile Training

- 1 day in-house Kanban Introduction
- Kanban Kickstart
  - 1 day Kanban Introduction, kickstart + follow up
- 1 day Scrum introduction
- Agile Review
- CSM, CSPO
- New: Kanban 2 day Training
  - March 8-9 Copenhagen
  - April week 16
In general

- Who are we?
- Let me know if you have questions
- Please help me remember the break 😊
- You Will Get the Slides
A COUPLE OF WAR STORIES
1: THE CLASSICAL KANBAN EXAMPLE, OPERATIONS
2: ORGANIZATIONAL RESISTANCE
3: SUBOPTIMIZATION
4: A MATURE AGILE TEAM
KANBAN PRINCIPLES
1. PRINCIPLE
VISUALIZE WORKFLOW
# Visualize Workflow

<table>
<thead>
<tr>
<th>Inbox</th>
<th>specification</th>
<th>Breakdown</th>
<th>Development</th>
<th>Code review</th>
<th>Test locally</th>
<th>Test on Staging</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Planned</td>
<td>In progress</td>
<td>Done</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Planned**: Details of the task in the planning stage.
- **In progress**: Details of the task during the development stage.
- **Done**: Details of the task after completion.

**Code review**

**Test locally**

**Test on Staging**

**Release**
2. PRINCIPLE
LIMIT WORK IN PROGRESS
WHAT IS WIP IN SOFTWARE?
## Limit Work in Progress

<table>
<thead>
<tr>
<th>Inbox</th>
<th>Specification</th>
<th>Breakdown</th>
<th>Development</th>
<th>Code review</th>
<th>Test locally</th>
<th>Test on Staging</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Planned</td>
<td>In progress</td>
<td>Done</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **WIP Limit = 2**
- **WIP Limit = 3**
Capacity Constrained System

- You can never overload a Kanban pull system!
3. PRINCIPLES: MAKE POLICIES EXPLICIT
## Make Explicit Policies Visual

<table>
<thead>
<tr>
<th>Inbox</th>
<th>Specification</th>
<th>Ready for Development</th>
<th>Development</th>
<th>Code review</th>
<th>Test locally</th>
<th>Test on Staging</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>In progr.</td>
<td>Done</td>
<td>Planned</td>
<td>In progr.</td>
<td>In progr.</td>
<td>Released:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>In progress</td>
<td>Done</td>
<td>Done</td>
<td>- Remove tickets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Write end date</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Review deploy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Update CFD,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Defect rate and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cycle time</td>
</tr>
</tbody>
</table>

- Only Core Functionality

**Write Start Date**

- Accept Criteria!

- Plan pairing

- Refactor

- TDD

- Cover: Unittest Int., Test Code Coverage Depl. issue

- Tester and Product Owner need 10 min. preparation

- Only Core Functionality
4. PRINCIPLE
MEASURE AND MANAGE FLOW
Cumulative Flow Diagrams

**How to Read a Cumulative Flow Diagram**

- **New Tasks**
- **Backlog Size**
- **Lead Time**
- **Cycle Time**
- **Work in Progress**
- **Remaining to be done**

*Legend:*
- Deployed
- Ready to Deploy
- Ready for Approval
- In Testing
- In Progress
- Ready to Start

*Supported by Kanban Triumph, Inc.*
WAIT TIME

MINUTES FROM THIS POINT

Ride a briskly-moving rotating carnical train through a wacky mickey to play the Toy Story characters’ favorite 3D games, passing to score points by shooting at targets with a Spring-Action Shooter.

CAUTION
Persons who are prone to motion sickness should not ride.

Supervise children at all times.
Wheelchair access

©JeanineY
5. PRINCIPLE IDENTIFY IMPROVEMENT OPPORTUNITIES
WHEN USED RIGHT POLICIES AND METRICS WILL DRIVE CHANGE
Identify Improvements

Inbox | Specification | Breakdown | Development | Code review | Test locally | Test on Staging | Release (Every Tuesday)

- Green: Planned
- Blue: In progress
- Purple: Done

Plan pairing

Remember: Unit test Int. test Coverage Depl. issue

Tester and PO need 10 min. preparation

Only Core functionality

Every Tuesday
"KANBAN IS LIKE GETTING A SHRINK FOR YOUR PROCESS"

- JAN OLOFSSON
Kanban Principles Overview

1. Visualize Workflow
2. Limit Work-In-Progress
3. Make Policies Explicit
4. Measure and Manage Flow
5. Identify Improvement Opportunities
REAL LIFE EXPERIENCES
FOCUSSING ON FLOW HELPS!
YOU WILL BECOME MORE AGILE
A "DRIVER" HELPS
YOU NEED BOTH THE "WHY" AND THE "HOW"
CHANGE MANAGEMENT IS STILL HARD 😊
UNCOACHED INITIATIVES FAIL!
QUALITY FOCUS HELPS. BUT YOU NEED PATIENCE
DON’T WORRY ABOUT A LACK OF FOCUS!
GETTING PEOPLE TO THINK ABOUT THE ENTIRE VALUE STREAM IS HARD!
Blocked Items
Don’t Forget the Vision!
DON’T WORRY ABOUT PEOPLE USING KANBAN AS AN EXCUSE TO REVERT TO FORMER PRACTICES
PEOPLE WILL COPY WHATEVER YOU SHOW THEM!
PLUG-IN AGILE STILL DOES NOT WORK!
Most Failures

- fall into one of the following categories
  – No management commitment
  – No crisis
  – Management wants it but does not live it
  – No autonomy
  – People focus only on the mechanics
DISCUSSIONS