The Scrum Gauntlet of Debt

An exercise/simulation that will illustrate to teams and managers why they need to complement Scrum with XP and what happens if you don't

Understand room make up

Show of hands :-

- Devs
- Testers
- Managers
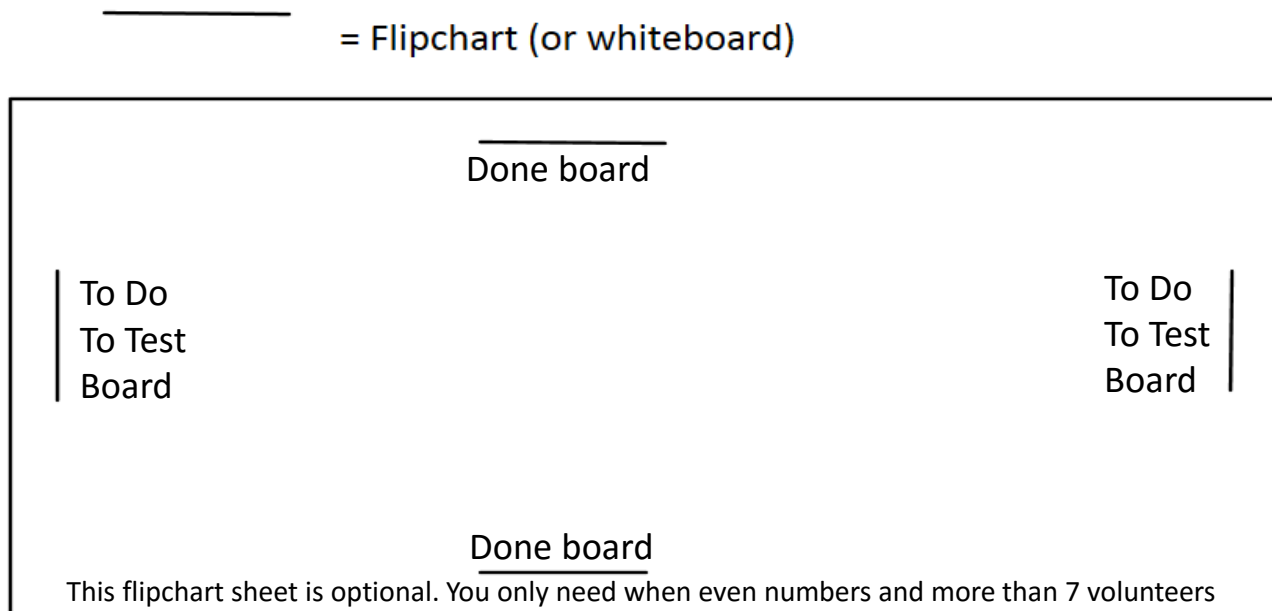- Scrum Masters
- Agile Consultants
- Other

Running the Gauntlet - Requirements

- Volunteers (7-12 volunteers)
- Two walls and two easels or 4 easels
- Chairs (observers can give up theirs while watching if need be)
- Sticky notes (small square size is better but any will work)
- Flipchart (Sticky is best)
- Blue tape
- Something with a countdown timer (smartphone apps work)
- Three surfaces for flipchart sheet. E.g. walls, whiteboard, easel
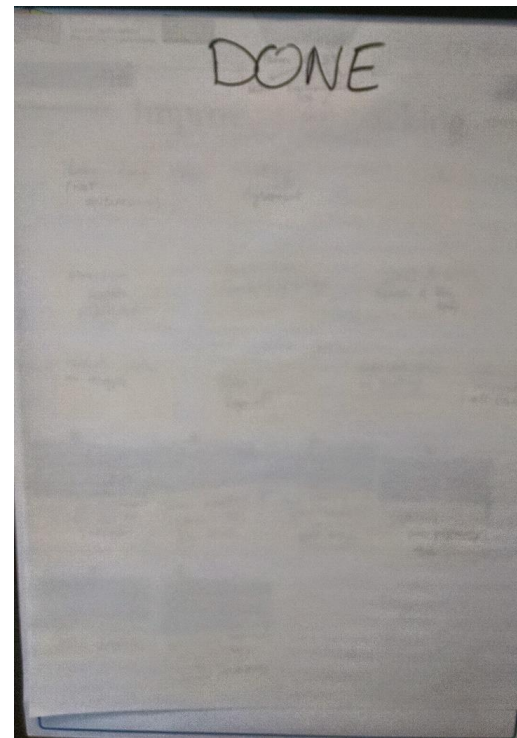- Marker pens to write on flipcharts (and whiteboard)
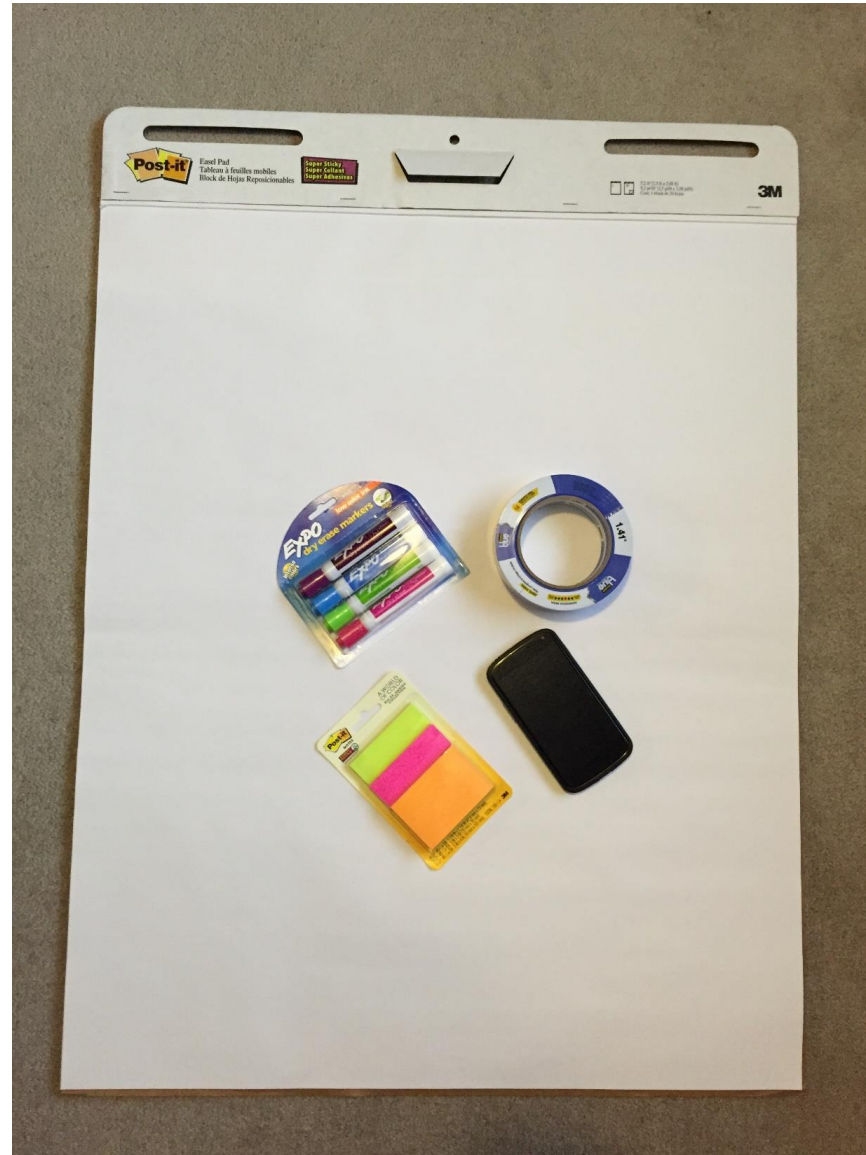
Prep – Create the Sprinting dev room

Clear a gauntlet area in the room and layout like this:

_____ = Flipchart (or whiteboard)

_____
Done board

To Do
To Test
Board

To Do
To Test
Board

Done board

This flipchart sheet is optional. You only need when even numbers and more than 7 volunteers

Create Backlog and Done Boards (Flipcharts)
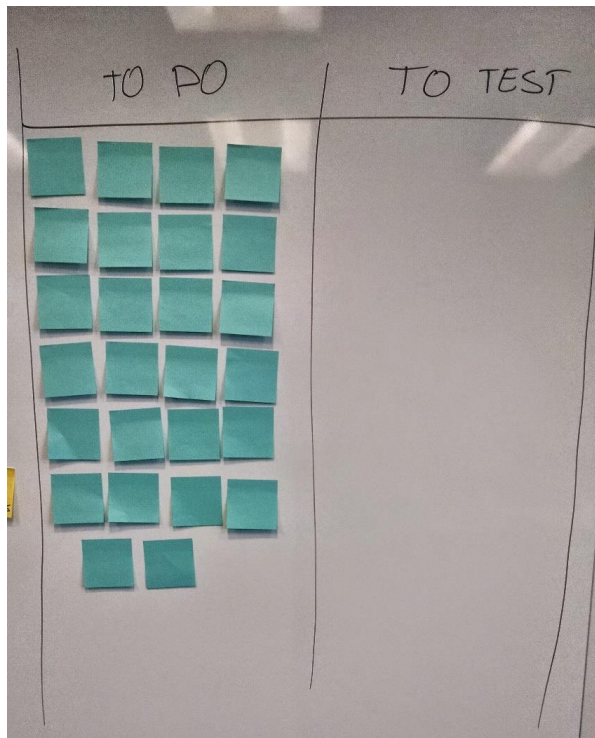
Tip

If using a flipchart, use the blue tape to hold the bottom corners of the flipchart down

Volunteers and Roles

Ask for volunteers with warning that it is a physical demonstration and there will be running. Do not volunteer unless comfortable with this.

Divide the volunteers into testers and developers using one of these formations depending on the number of volunteers

- 10 devs and 1or 2 testers

- 8 devs and 1 or 2 testers

- 6 devs and 1 or 2 testers

Divide devs into two even groups

Prep participants and Timer Role

- Identify another timer volunteer (who has a smart device with countdown timer app)

- Explain to participants that this is a simulation. The goal is not to "win" and game the system. The goal is to mimic what real life looks like

- Instruct participants to act like a dysfunctional team and not communicate with each other (otherwise they start gaming the system)

Placing of volunteers in the gauntlet

Divide devs in two groups and place volunteers like so :

_____ = Flipchart (or whiteboard)

_____
Tester

Devs | Devs

Optional Tester (if even numbers)
_____
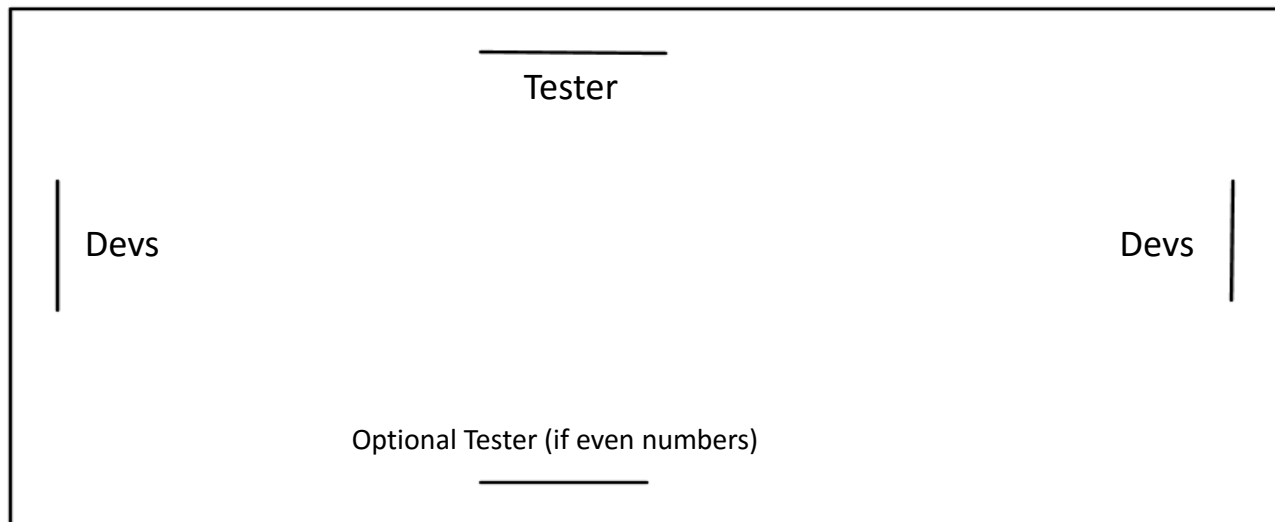
Rules of the game/simulation

- An iteration lasts 30 seconds

- Developers run to opposite wall, pick a story token (sticky note) from TODO and run back to their starting wall, placing the token in the TO TEST column

- A Dev can only work on one story at a time

- Testers wait by their Done board until they see stories ready to TEST and then run to (either) board, grab up to 3 stories (but no more than 3) and return them to their done board

- Everyone must run (That is why they call it Sprinting!)

- No collusion or working out a system. You are mimicking a dysfunctional team (Be sure to have explained this or else they game the system)

- BE CAREFUL!!! (Re-iterate and ask volunteers again if comfortable)

BE CAREFUL!!!

- Health and safety
- Don't have anyone pregnant, wearing high heels or with physical disabilities that may cause risk to themselves or others in the volunteers
- Make sure volunteers are comfortable with exercise otherwise switch out
- Re-iterate to volunteers Safety First!
- Be careful!!!

Any Questions? Then Go!

Notes for facilitator – Sprint 0

- While the team are sprinting, you play the role of a manager and shout at the team "FASTER", "I Need more features" etc.

- Run the first sprint without keeping score (it's a test run)

- (the reason to not keep score is that because the velocity of the first one is usually lower because they are working it out and it gives you a chance to find out if anyone misunderstood anything)

- At the end of the sprint tell the team to reset the boards

- Ask them if they are warmed up and a little puffed? "That is why they call it a sprint!"

Management Abuses to shout at team

- Faster!
- I need more features!
- What is wrong with this team?
- Why are they so slow?
- Why are your estimates all wrong?
- More velocity!
- Development monkeys!
- Screw code practices – I need Features!
- Stop that pair programming nonsense!
- My last team was better than you guys
- Don't make me fire you
- I should out source the lot of you
- Etc…

Sprint 1 & introduction of technical debt

- Keep shouting abuse

- This sprint, record the velocity
- Now, place chairs in the gauntlet/sprinting area with the explanation "These chairs represent technical debt. Every story you completed in the last iteration, you left behind technical debt. This is what your codebase starts to look like. Can anyone relate? Do your teams have technical debt" (I've yet to have a no answer)
- Be even more careful!!!! No leaping or moving chairs.
- Testers will move one task from TO TEST to TO DO on each run (to represent an increase in bug count due to technical debt)
- Go! Start sprint 2 and be sure to keep the abuse

Sprint 2 Management Abuse

- I don't want to hear about technical debt excuses – I want features damn it!
- When can we release?
- No refactoring or cleaning up – Features!
- Run, run, run!
- Lazy coders!
- I'm paying you too much!
- There's going to be overtime!

Sprint 3 & introduction of refactoring

- Put more chairs in the gauntlet to make it near impossible for the team
- Explain to the room that this is what a day looks like for a developer in a legacy system
- This is what their life is like – get confirmation from the room e.g. "can anyone relate?"
- Be sure to point out the team are looking haggard and tired at this point
- Ask them if they think they can keep this pace up indefinitely and what they think of the upcoming sprint!

- Now introduce refactoring and sustainable pace
- Developers are now required to walk and not run (agile principle of sustainable pace)
- Developers are required to clean up after themselves and move one chair out of the way on their return leg. One chair per story

- Keep shouting abuse

Sprint 3 Management Abuse

- What the hell is going on?
- Who said you guys could slow down?
- Stop this refactoring nonsense!
- Run damn it, run!
- I didn't hire extreme programmers – I hired scrum developers
- I want speed, not quality
- Why are you pair programming?
- There goes our velocity!

Post Exercise discussions

The physical portion is now complete. Ask participants to be seated.

Your recording of velocity should look something like this..

Sprint #1 – 13
Sprint #2 – 10
Sprint #3 – 8

Bring this sheet of flipchart to the front of the room. You now begin some discussions…

fast-agile.com