Scrumban Game

Introduction

The Scrumban Game is a fun way to practice evolutionary change in the context of Agile software development with Scrum. The game simulates the notorious and first ever Kanban case study about the XIT team at Microsoft.

In the game you will successfully improve the performance of your Scrum team by applying the Kanban method, while keeping the core Scrum process intact.

Here are all the file you need to play the game
https://github.com/dimiterbak/scrumban_game

Context

Your Scrum team maintains and improves an existing ERP system in production.

- The Scrum team works on bugs and small new features called Change Requests (CR) and Production Text Changes (PTC).
- There are three Product Owners represent three corporate departments.
- The Scrum Master is one of the developers.
- Sprints are 10 working days long.
- All finished work can be released immediately thanks to CI/CD.

In the game you are hired as a coach to help a Scrum team to improve. To achieve success, you have to improve the performance of the team in terms of Customer lead time and Throughput.

As Kanban method practitioner you decide to introduce and apply the six Kanban practices, namely:

- CP1: Visualize
- CP2: Limit work-in-progress (WIP)
- CP3: Manage flow
- CP4: Make policies explicit
- CP5: Implement feedback loops
- CP6: Improve collaboratively, evolve experimentally (using models and the scientific method)

Improvements

Your improvement efforts will go through three sprints:

1) In the first sprint your goal is the Scrum team to understand and measure what is the current state of their system.
2) In the second sprint you advice the team to stop doing estimates and for the CRs offer a Service Level Expectation (SLE) instead.
3) In the third sprint the team will introduce WIP limits and switch to on-demand replenishment of the Sprint backlog.

Learning outcomes

Playing the game you will learn:

- How to prepare and make use of Cumulative Flow Diagram (CFD), Customer Lead Time histogram, System Lead Time histogram, Flow Efficiency, Due Date Performance, SLE performance.
- How to offer Classes of Service i.e. Service Level Expectation (SLE)
- How to manage the flow of work in your Scrum team
- How to visualize your work
- How to limit work-in-progress (WIP)
- How to make policies explicit
- How to Implement feedback loops
- How to Improve collaboratively, evolve experimentally

Rules of the game

The game has rules and policies. Rules are fixed and cannot be changed. Policies are changeable.

Visualizing the work of the Scrum team

Teams visualize their work using a kanban board. Below is the board for the first sprint.

Each team has 10 days to finish 10 user stories. Th team follows the standard Sprint workflow:

- Plan the Sprint
  - Populate the Sprint Backlog
- Execute the Sprint
- Run a Sprint Review

- Run a Sprint Retrospective

Work types

- Change request (CR). CRs need both development and testing!
- "Rough order of magnitude" (ROM) estimate
- Production Text Change (PTC)

When **during the sprint** a new change request (CR) arrives it is sent to the Scrum Team for a "rough order of magnitude" (ROM) estimate.

ROM estimates are essential to facilitate both budgeting and prioritization.

All requests for ROM should be fulfilled per the Service Level expectation (SLE) of 2 days.

ROMs require only development time and no testing.

Production Text Changes (PTC) Need only testing, no development. PTCs have a fixed delivery date.

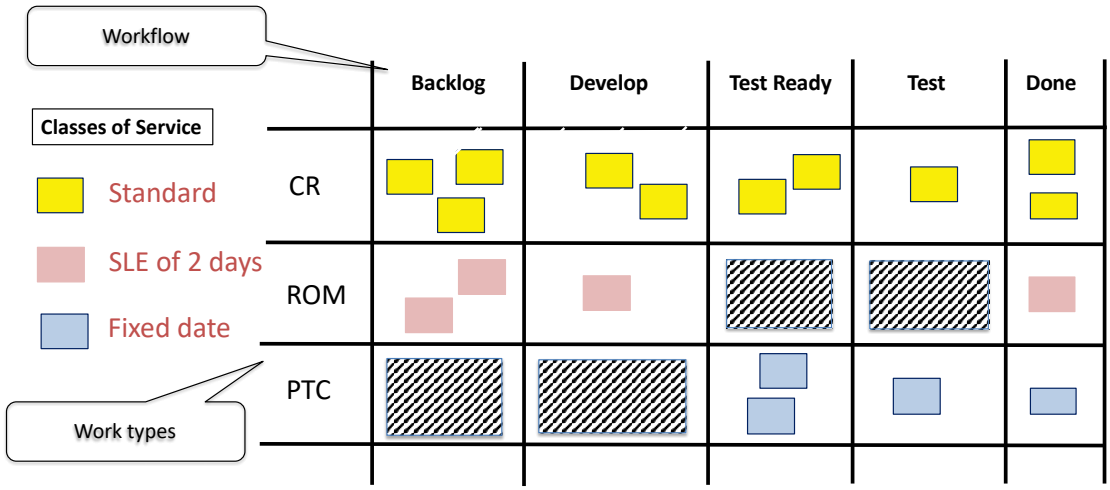
Sprint goal is to deliver all CRs, PTCs and ROMs!

Classes of Service

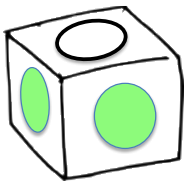
Standard	Standard (to be delivered before the end of the Sprint)
SLE	Service Level Expectation (SLE)
Fixed Date	To be delivered at a fixed day of the sprint

On the kanban board we have visualized:

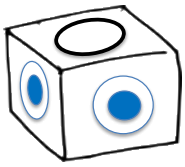
- work types as swim-lanes (CR, ROM, PTC)
- the workflow (knowledge discovery process) of each work type
- classes of service (Standard, SLE, Fixed date)



A die for each role

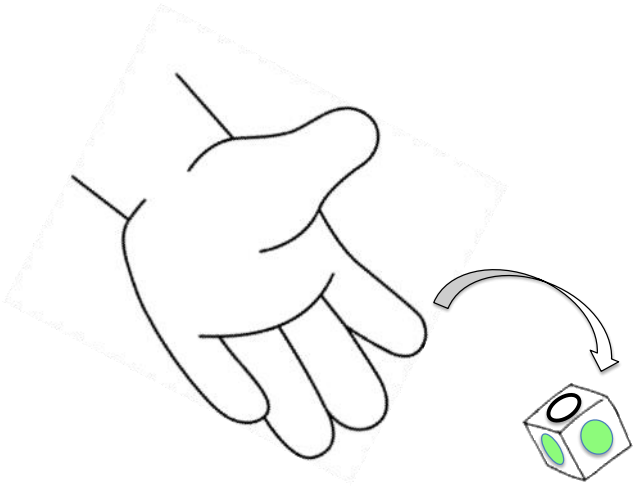


Dev Team member's die



Product Owner's die

Roll a die



Rule for developing a new work item

If you roll White	If you roll Green
Do one of: • Advance one of the work items to the next column to the right • **OR** help others	• Leave the card in it's current column

Rule for converting a ROM into a new CR

When the ROM card enters Done column roll the Product Owner die.

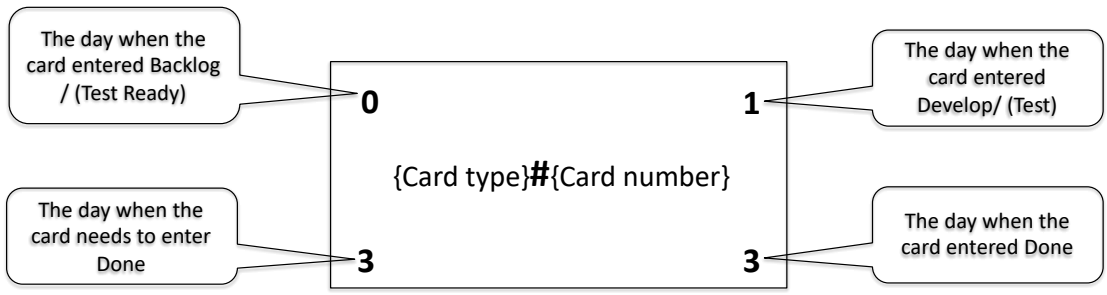
If you roll White	If you roll Blue
• create a new CR card in the Backlog and put the day in the top left corner.	• Store the card in the Trash bin.

In case of new work from the events cards

- Generate:
  - The requested number of ROMs of color pink and put them in the Backlog
  - The requested number of PTCs of color blue and put them in "Test Ready"
- Write the number of work item in the center of a sticky note with PTC/ROM leaving room top and bottom.
- Write the requested delivery day in the bottom left corner
- Put the current sprit day in the top left corner.

Metrics

Tracking work



Illegal Moves

When you pair to help a teammate both of you are allowed to move only one card.

A card can move only once per day.

Setup (for each period)

1. All participants to get into teams of 2 - 4 people.
2. Each team to generate a backlog of 20 features using post-it notes of size 51x38 mm.

   Write the number of each feature in the center of a sticky note, leaving room top and bottom. Leave the cards in the “Ready” column.

Timing

It takes between 60 and 90 minutes to go through all three sprints.

Materials and downloads

Materials for one team:

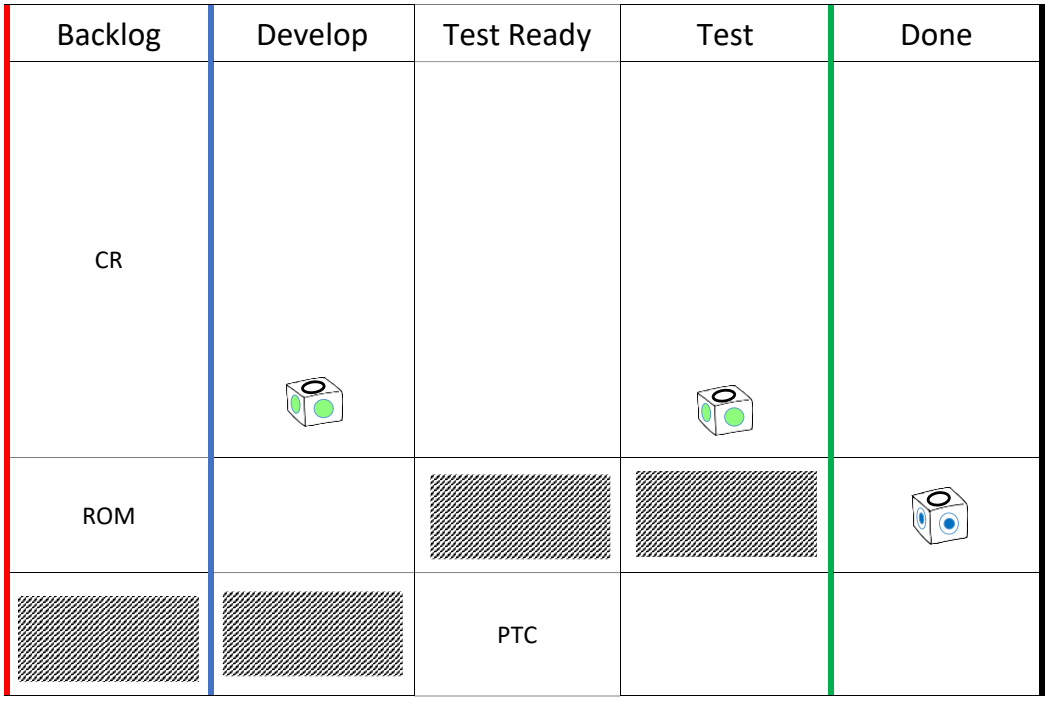
- Scrumban Game Board_S1 v5 (A2) - 1 piece
- Scrumban Game Board_S2 v5 (A2) - 1 piece
- Scrumban Game CFD_v6 (A3) - 1 piece
- Scrumban Game Hist_DD_v2 (A3) - 1 piece
- Scrumban Game CR_CLT v1 (A3) - 1 piece
- Scrumban Game ROM_SLE v6 (A3) - 1 piece
- Scrumban Event Cards_v4 (A4) - 1 piece
- 40 small (51mm x 38mm) sticky notes of color pink.
- 30 small (51mm x 38mm) sticky notes of color blue.
- 20 small (51mm x 38mm) sticky notes of color yellow.
- Four pens of colors black, green, blue, red.
- One (4white/2blue) die
- Five (4white/2green) dice

Roles

- **Development Team member.** Manages the Sprint Backlog, keeps statistics, plans the sprint, starts, moves and completes user stories.
- **Product Owner.** Manages the Product Backlog.
- **Scrum Master.** One of the developers.

Sprints

Sprint #1: Visualization and measurement



Sprint planning

What can be delivered?

- In the "Backlog" column generate:
  - 10 CRs of color yellow.
- Write "CR" and the number of work item in the center of a sticky note. The number is continuous – from 1 to 10.
- Put 0 in the top left corner leaving room top and bottom.

The backlog is prioritized – deliver accordingly!

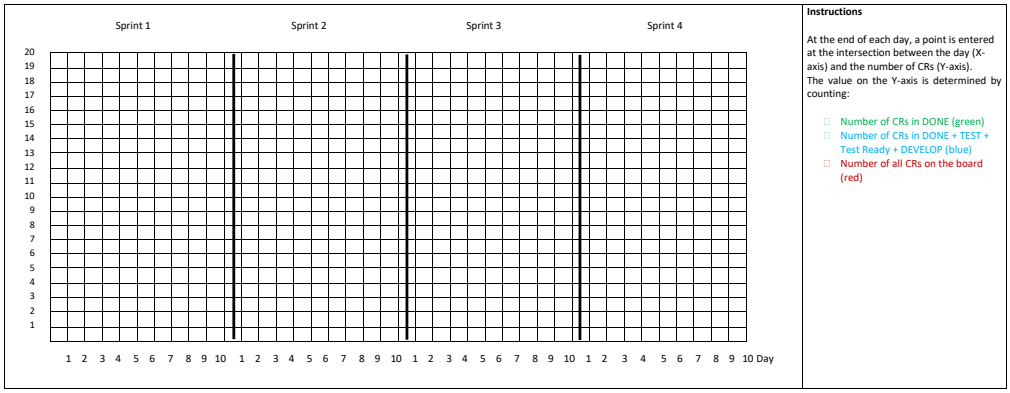
Sprint execution

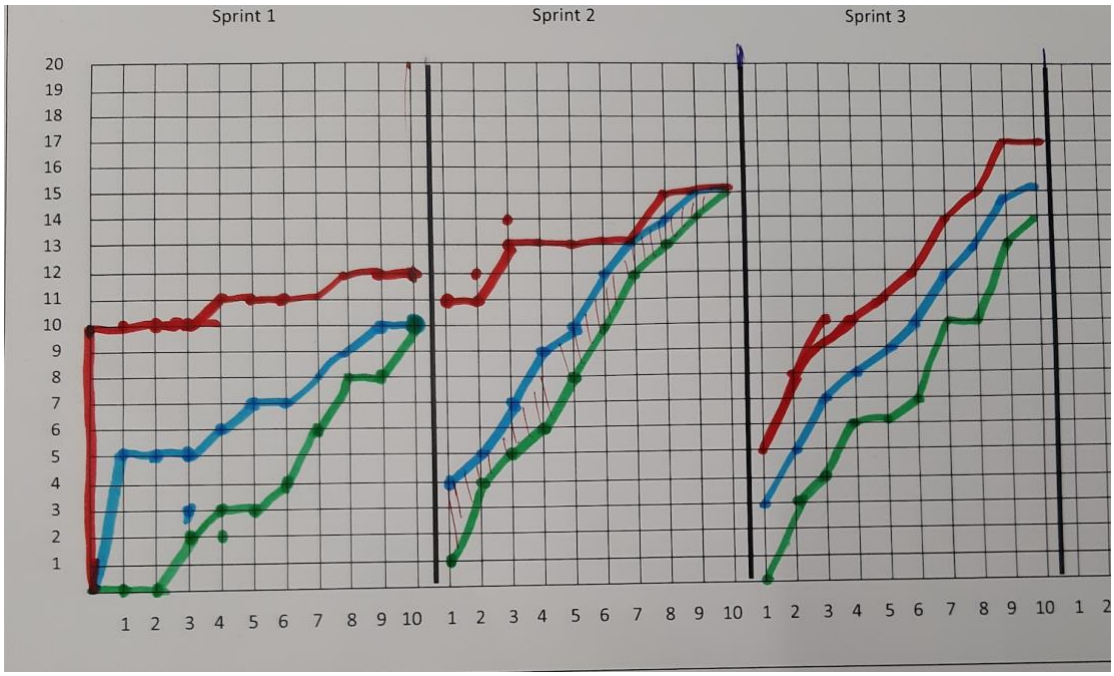
Daily routine of the Sprint:

1. Pick an event card and follow its instructions.
2. Each of the Dev team members should:
   a. Move the card they will work on in the respective column **with a die**
   b. Roll a die to see if the card can move to the next column to the right
   c. share with the other team members the result, then move the work items on the Kanban board according to the rules and the policies. Please don't think that the team members should take turns to move.

3. At the end of the day the Dev team collects CFD for the CRs using the provided sheet.

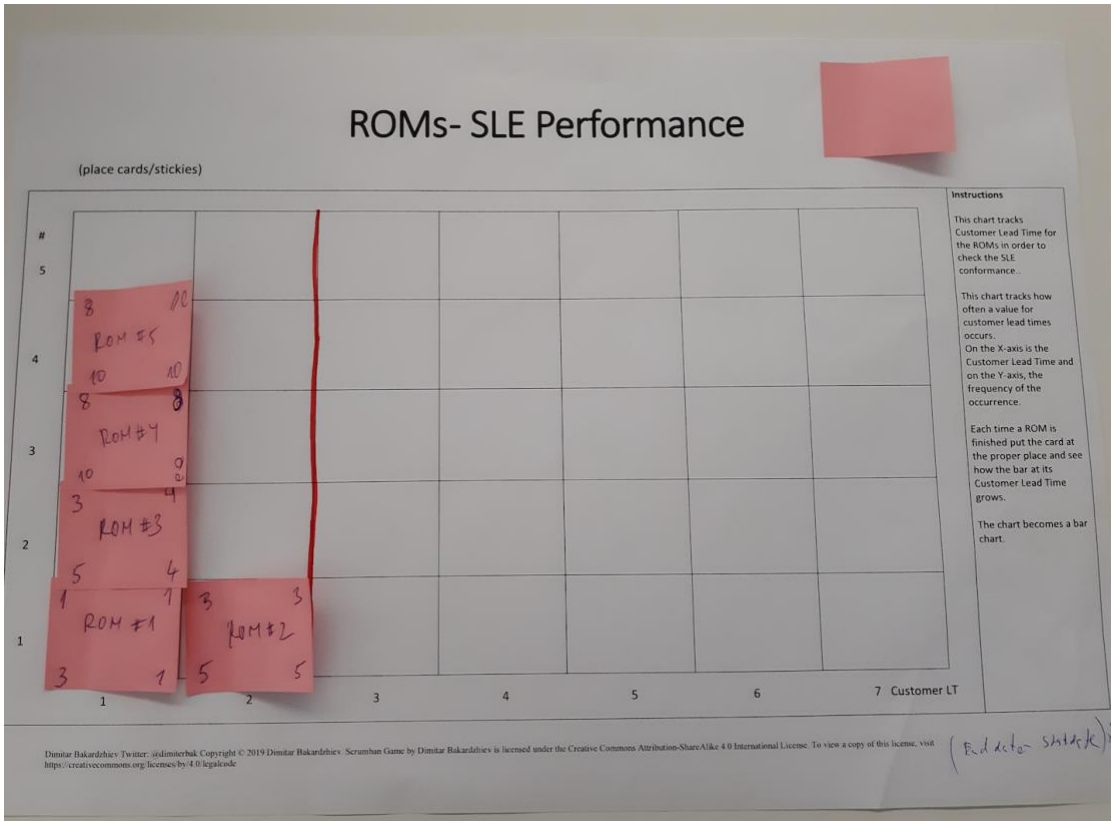
Cumulative Flow Diagram (CFD) for the Change Requests (CR)

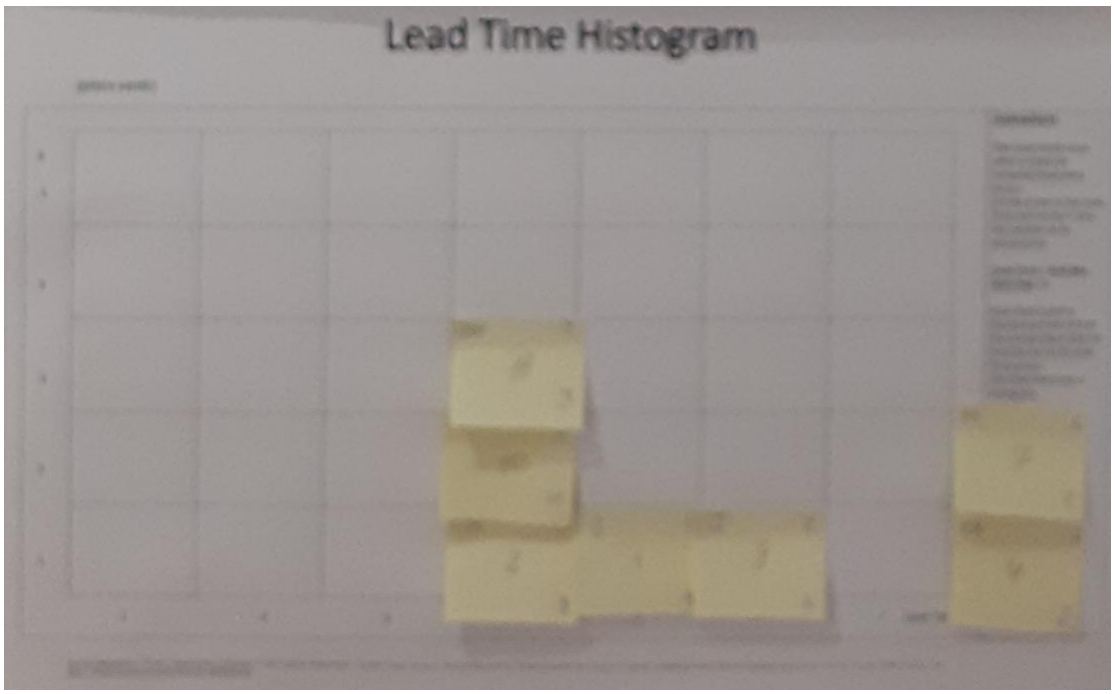


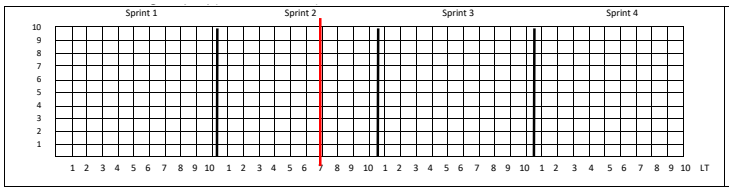


Sprint review

End of the Sprint: collect statistics!


ROMs- SLE Performance
(place cards/stickies)
#
5
4
3
2
1
ROM #5
ROM #4
ROM #3
ROM #1
ROM #2
1
2
3
4
5
6
7 Customer LT
Instructions
This chart tracks Customer Lead Time for the ROMs in order to check the SLE conformance.
This chart tracks how often a value for customer lead times occurs.
On the X-axis is the Customer Lead Time and on the Y-axis, the frequency of the occurrence.
Each time a ROM is finished put the card at the proper place and see how the bar at its Customer Lead Time grows.
The chart becomes a bar chart.
Dimitar Bakardzhiev Twitter: @dimiterbak Copyright © 2019 Dimitar Bakardzhiev. Scrumban Game by Dimitar Bakardzhiev is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/legalcode


Lead Time Histogram



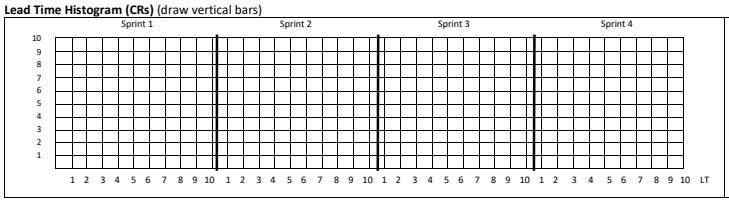
Instructions

This chart tracks how often a value for customer lead times occur.
On the X-axis is the Customer Lead Time and on the Y-axis, the frequency of the occurrence.

Customer Lead Time =
The day CR finished
–
The day CR was requested

If 0 or 1 then take 1

Lead Time Histogram (CRs) (draw vertical bars)



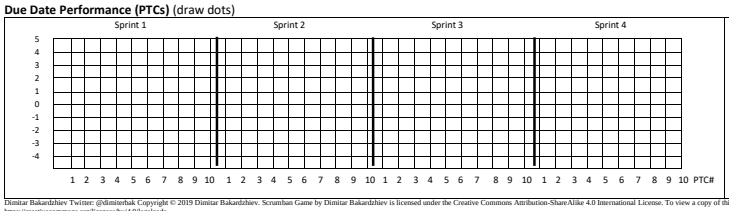
Instructions

This chart tracks how often a value for lead times occur.
On the X-axis is the Lead Time and on the Y-axis, the frequency of the occurrence.

Lead Time =
The day CR finished
–
The day CR entered Develop

If 0 or 1 then take 1

Due Date Performance (PTCs) (draw dots)

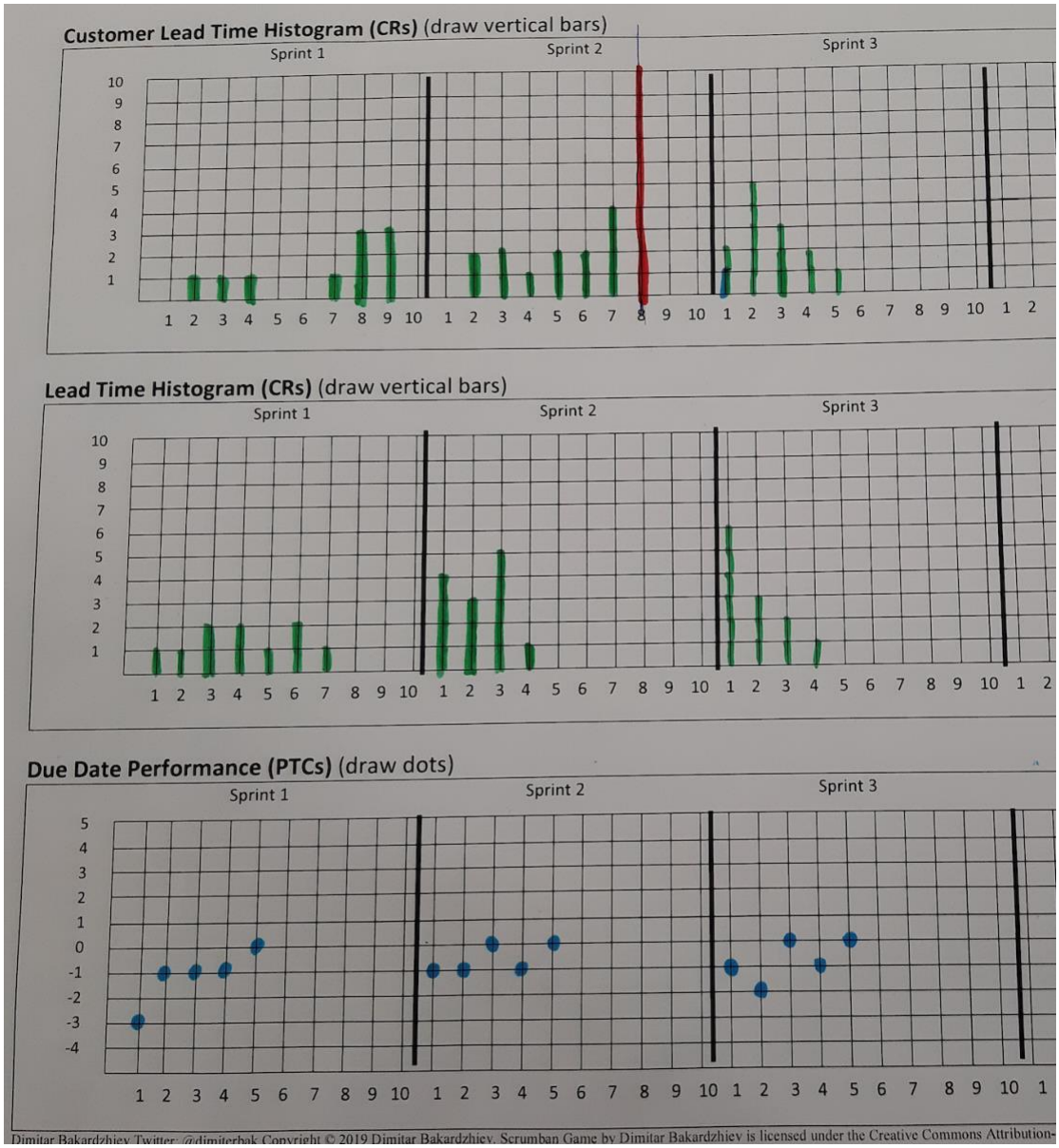


Instructions

This chart tracks due date performance for the PTCs.
On the X-axis is the number of PTC card and on the Y-axis is the difference between its delivery day and its due date.

Due day- Done day
If 0 or 1 then take 1

E.g. if a PTC is delivered on day 6 and its due date was 5 then the difference is 6-5=1 day overdue..



Sprint retrospective

Discuss how fit is the team for its purpose?
Check what the data reveals:

- CRs Customer lead time histogram
- CRs throughput (from CFD)
- PTCs Due Date performance
- ROMs SLE performance

Sources of dissatisfaction:

Internal (Development team)	**External (Product Owners)**
Constantly interrupted to make ROM estimates	Small CRs take too long to develop
Estimates are expected to be highly accurate and take time to develop	Delivery is completely unpredictable
Testers occasionally swamped by PTCs and cannot respond to developers	Constantly breaking promises

Analysis results ROM is non-value-added activity from CR Lead time perspective!

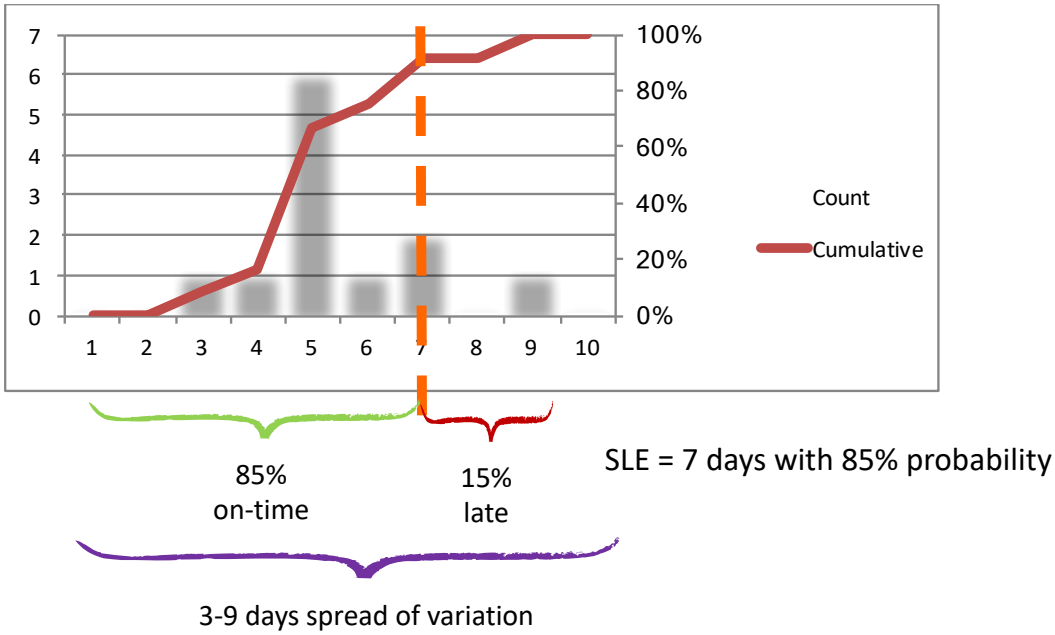
Changes

Offer to drop making estimates in return for an SLE for delivering a CR.

- Stop doing ROMs - thus free capacity for Scrum Team.
- Offer a SLE for the CRs instead based on the historical data. This way the Product Owners will be able to continue doing their prioritization.
- Commit to deliver all the CRs!

Establish your SLE for CRs: how many days with what probability?

Use Customer Lead Time Distribution to establish SLE for CRs



Debrief

In your teams, list and prepare to report back:

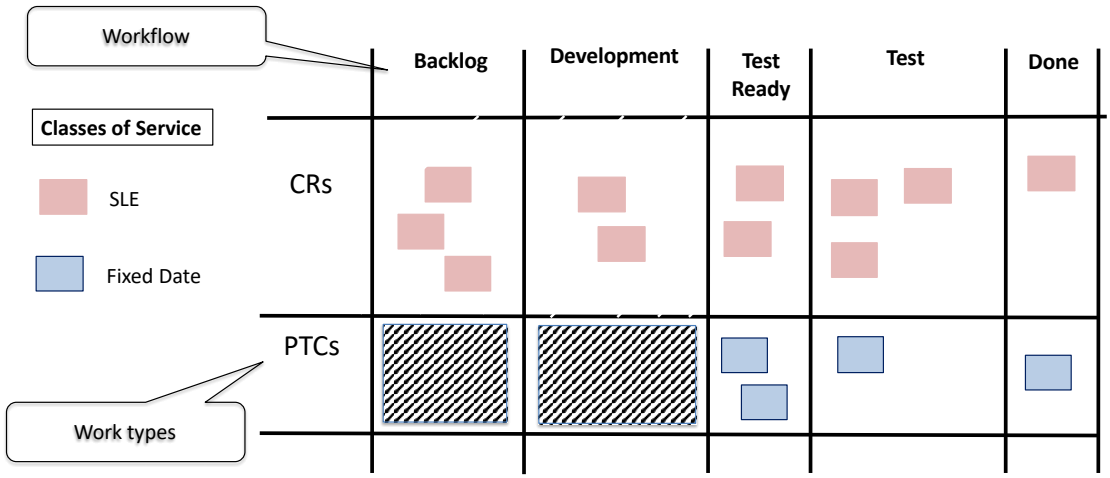
- Concepts, practices, and outcomes simulated in the game
- Your observations
- Workplace parallels

Confirm the application of Kanban Method's three core practices:

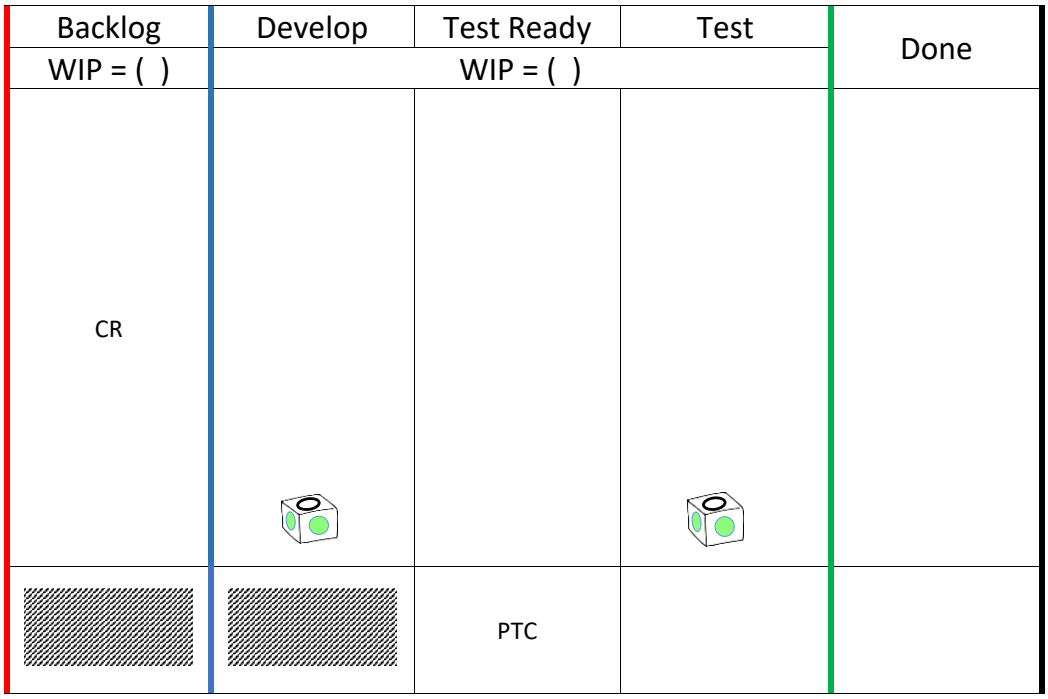
- CP1: Visualize
  - Work items
  - Work flow
  - Work item state – where in the work flow, whether blocked
- CP4: Make policies explicit
- CP5: Implement feedback loops
  - Daily Scrum meeting

Sprint #2: Drop doing ROMs and offer an SLE for the CRs

Change the kanban board:



Use the bellow file:



Sprint planning

What can be delivered?

- In the “Backlog” column generate:
  - 10 CRs of color yellow.
- Write “CR” and the number of work item in the center of a sticky note. The number is continuous – from 1 to 10.
- Put 0 in the top left corner leaving room top and bottom.

The backlog is prioritized – deliver accordingly!

Sprint execution

Daily routine of the Sprint:

1. Pick an event card and follow its instructions. If there is a ROM – create a new CR instead!
2. Each of the Dev team members should:
   a. Move the card they will work on in the respective column **with a die**
   b. Roll a die to see if the card can move to the next column to the right
   c. Share with the other team members the result, then move the work items on the Kanban board according to the rules and the policies. Please don’t think that the team members should take turns to move.
3. At the end of the day the Dev team collects CFD for the CRs.

Sprint review

End of the Sprint: collect statistics!

Sprint retrospective

Discuss how fit is the team for its purpose?
Check what the data reveals:

- Customer lead time SLE performance
- CRs throughput
- PTCs Due Date performance
- Lead time(Engineering effort) histogram

Sources of dissatisfaction:

Internal (Development team)	**External (Product Owners)**
Testers occasionally swamped by PTCs and cannot respond to developers	Small CRs take too long to develop

Analysis results

- There are two queueing systems – Product Owners and Dev Team.
- Both queueing systems are loosely-coupled (not one kanban system).
- There is a gap between the CR commitment point and Development start point.

Changes

- Limit the Backlog size to two CRs. Thus create a kanban system from the two decoupled systems: Product Owners and Scrum Team.
    - Transfer the responsibility for the decision "What to work on next?" where it belongs to – the Product Owners.
    - Remove the gap between the CR commitment point and Development start point by making the Scrum Team commit to a delivery date only when a CR is actually picked by the Product Owners to be worked on next.
- Limit Doing area size to three CRs.
- Replenish on-demand the now limited Backlog. Thus shrinks the difference between Customer and Development lead times.

Debrief

Discuss and prepare to report back:

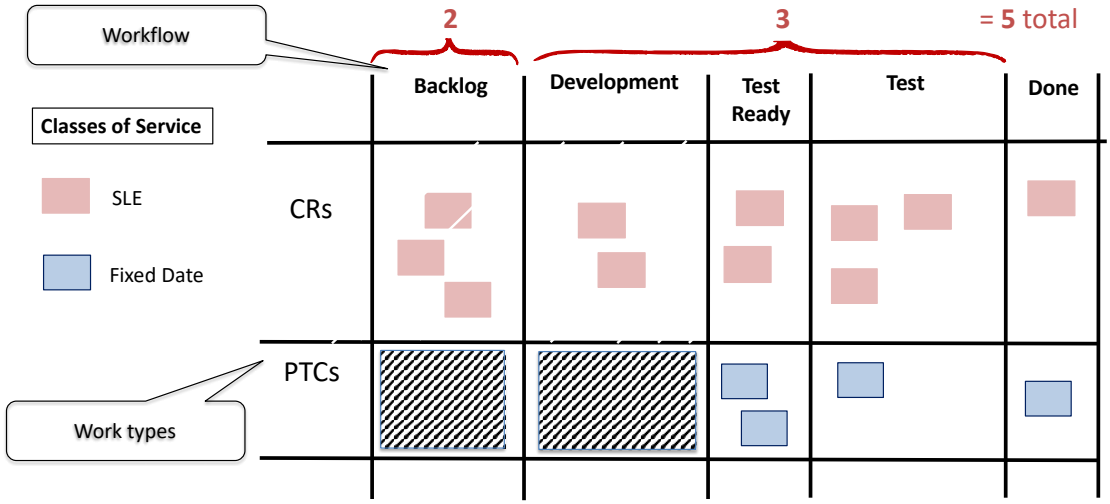
- What changed?
    - Benefits?
    - Drawbacks?
- Workplace parallels

Confirm the application of the two core practices:

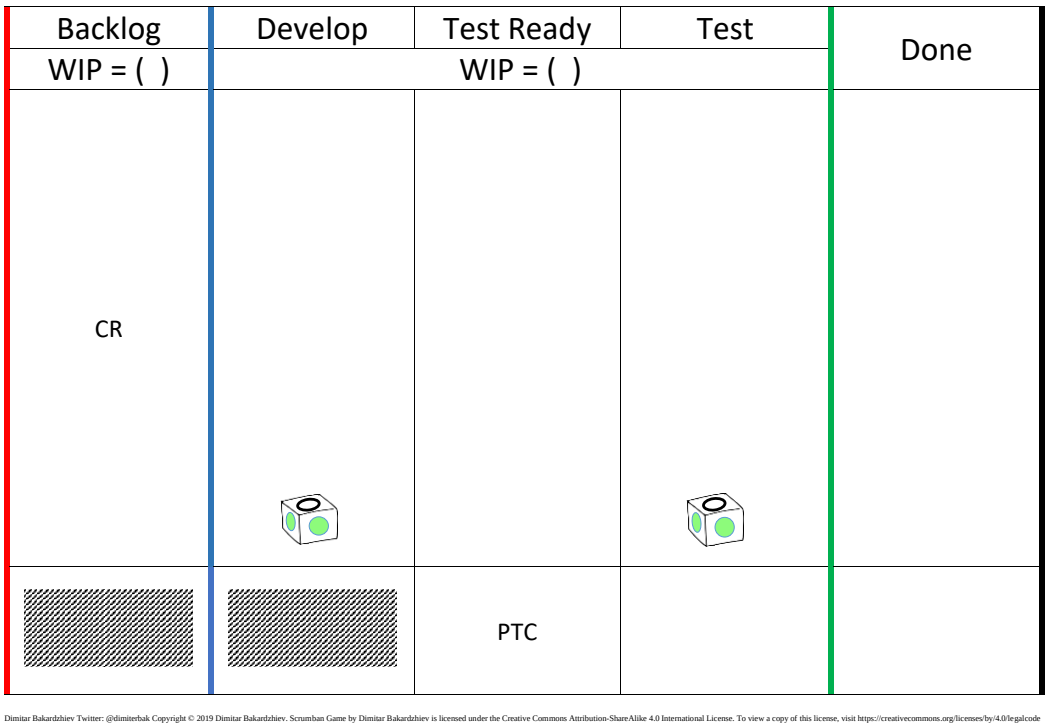
- CP3: Manage flow
    - Flow (smoothness, timeliness, economic outcomes)

Sprint #3: Limit WIP

New kanban board



Use the bellow file with WIP limits set:



Sprint planning

- In the “Backlog” column generate:
  - 2 CRs of color yellow.

- Write “CR” and the number of work item in the center of a sticky note. The number is continuous – from 1 to 2.
- Put 0 in the top left corner leaving room top and bottom.

The backlog is prioritized – deliver accordingly!

On-demand Replenishment:
- CRs get into Backlog on demand!
- Add the new CRs at the end of each day.
- Write “CR” and the number of work item in the center of a sticky note. The number is continuous – from 3 to 10.
- Put 0 in the top left corner leaving room top and bottom.

Sprint execution

Daily routine of the Sprint:

1. Pick an event card and follow its instructions. If there is a ROM – ignore it!
2. Each of the Dev team members should:
    a. Move the card they will work on in the respective column **with a die**
    b. Roll a die to see if the card can move to the next column to the right
    c. Share with the other team members the result, then move the work items on the Kanban board according to the rules and the policies. Please don’t think that the team members should take turns to move.
3. Each team member moves the cards on the board according to the rules.
4. At the end of the day the Dev team:
    a. collects CFD for the CRs.
    b. If there are free slots in Backlog then the Dev team would replenish Backlog with new CRs.

Sprint review

End of the Sprint: collect statistics!

Sprint retrospective

Discuss how fit is the team for its purpose?
Check what the data reveals:
- Customer lead time SLE performance
- CRs throughput
- PTCs Due Date performance
- Lead time(Engineering effort) histogram

Debrief

Discuss and prepare to report back:

- What changed?
    - Benefits?
    - Drawbacks?
- Workplace parallels

Confirm the application of the last core practice:

- CP2: Limit work-in-progress (WIP)
    - Column limits, one way to balance workload vs capacity
    - We made a true kanban system (pull and limiting WIP)
- CP6: Improve collaboratively, evolve experimentally (using models and the scientific method)
    - We created conditions for collaboration in delivery
    - Nothing collaborative, experimental or scientific about our change

Final debrief

After Sprint #3 is finished hold a final debrief. Ask all teams playing the game to share their results and reflect on them

Facilitation information

- When debriefing Sprint #1 use phrases such as “Being busy is not the same as being productive”, “All starting and no finishing” and “Lots of activity, little delivery”.
- Ensure everyone agrees before playing Sprint #3 that the only change has been the introduction of WIP limits. It is only a policy change – emphasize on that if needed.
- After Sprint #3 is finished stick the statistics papers on the wall. Use that for the final debrief.

Customization