Kanbanize the Release Engineering Process

Noureddine Kerzazi (nkerzazi@gmail.com)

Release Manager at Payza.com

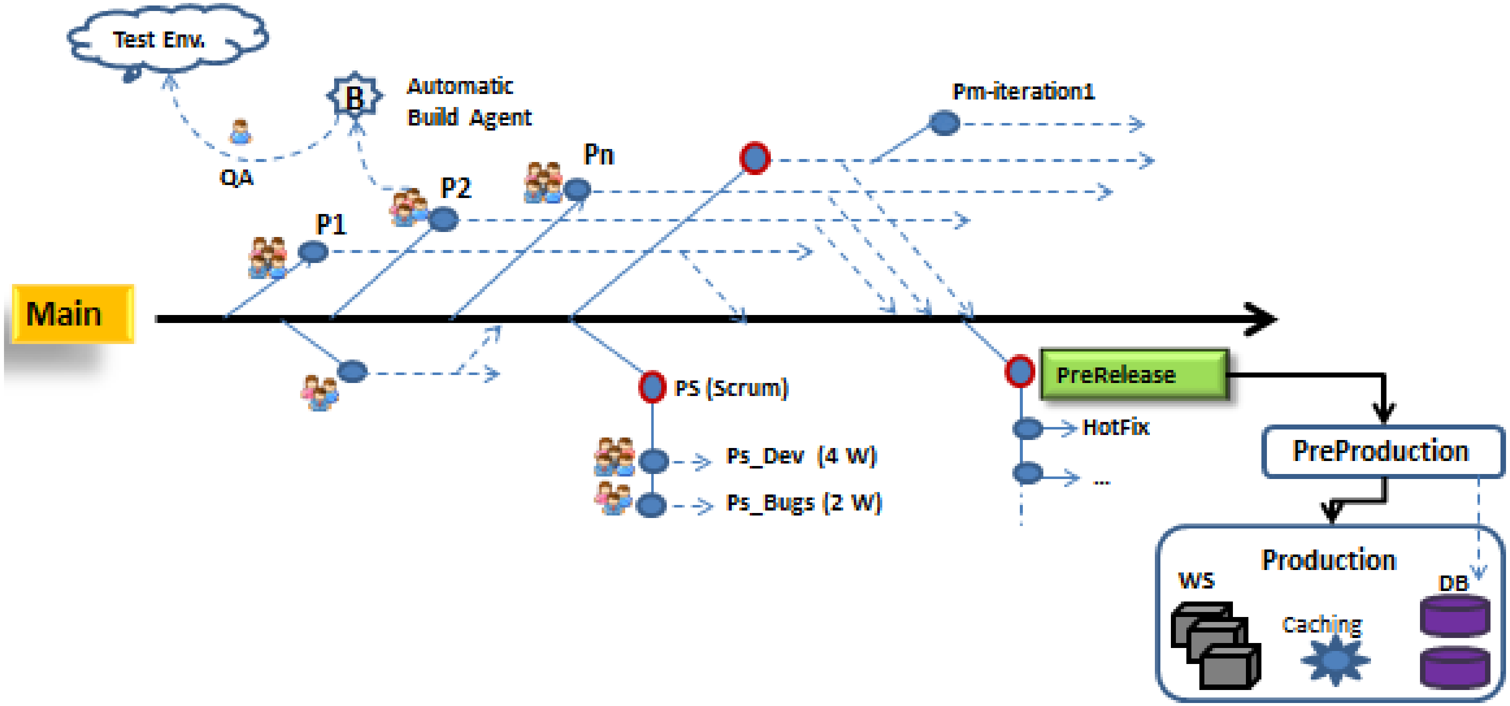
&

Post-doctoral researcher within the NSERC Fellowship program

Agenda

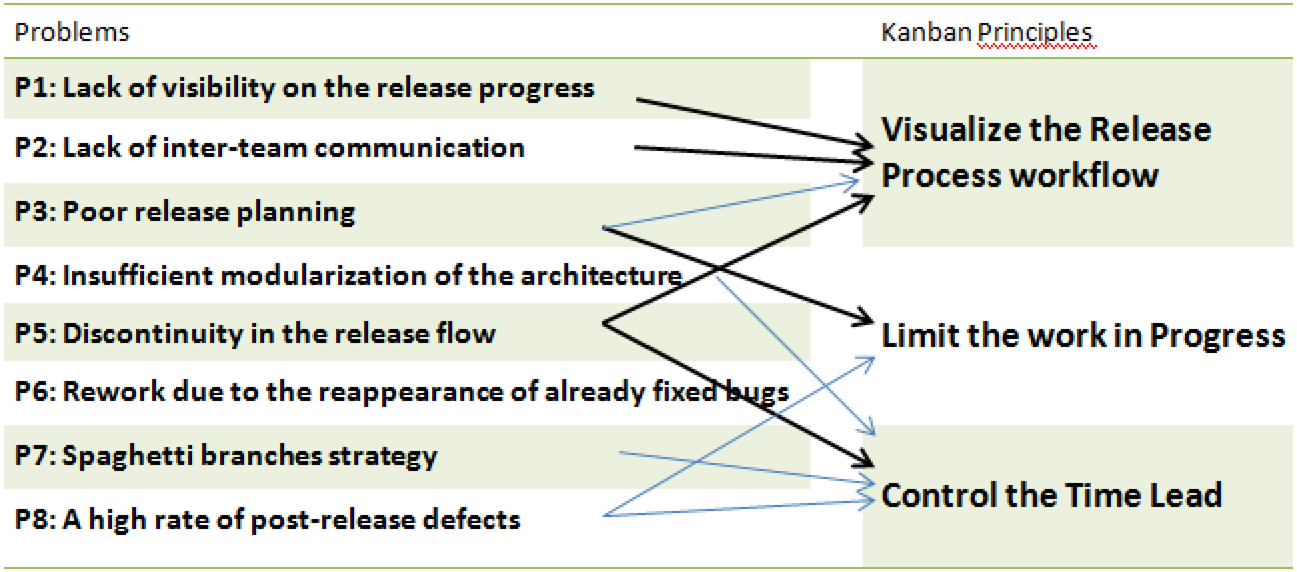
- Context and Motivations
- Pre-Transition
  - Modeling the Release Process
  - Principles of Kanban
- Transition
  - Approach
  - Implementation of the tool
- Post-Transition
  - Evaluation on the Kanban Transition
  - Practical Implications
- Conclusions and Outlooks

Context



- System composed of over than 1.5 million lines of code
  - Organized in more than 8524 source code files;
  - One solution structured around 9 Projects (Main, back office, Mobile, API, Sandbox, etc.);
  - Technically, one web-based solution with 46 projects;
- More than 100 persons involved in the system (Devs, BAs, Testers, Architects, DBAs, etc.).
- More than 70 branches in the source control system to allow parallel development.

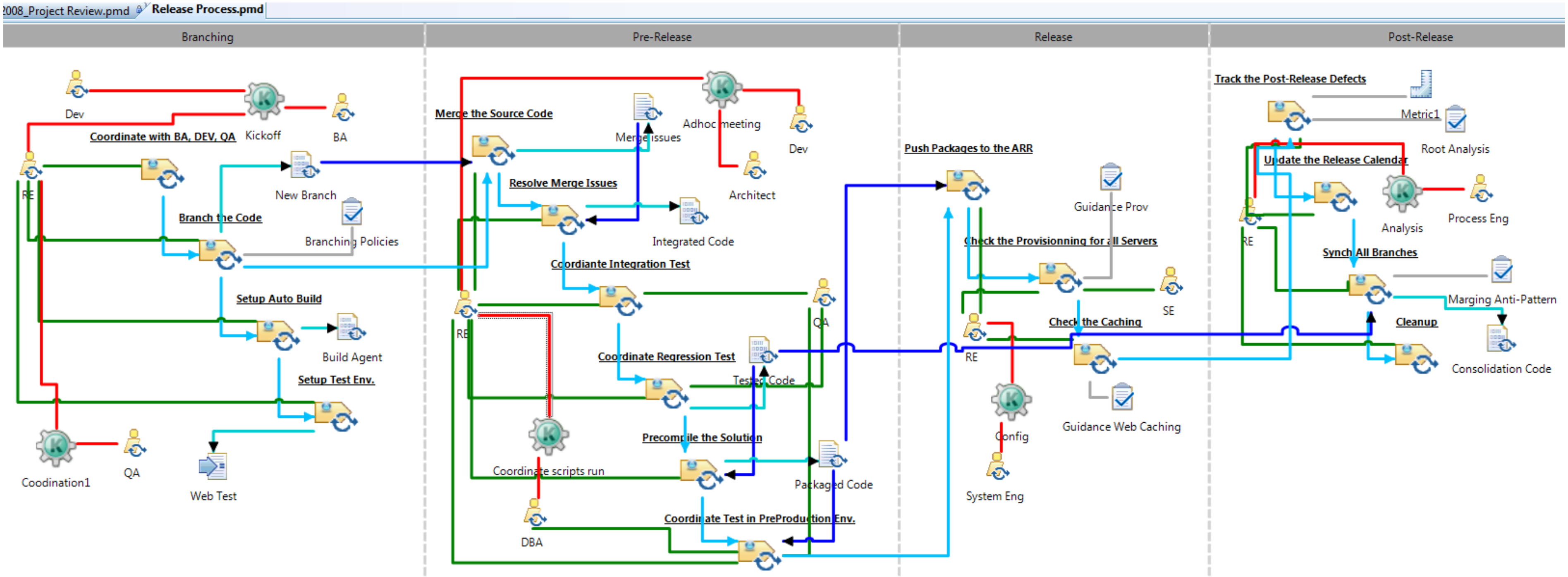
Motivations



Problems	Kanban Principles
P1: Lack of visibility on the release progress	Visualize the Release Process workflow
P2: Lack of inter-team communication	
P3: Poor release planning	
P4: Insufficient modularization of the architecture	
P5: Discontinuity in the release flow	Limit the work in Progress
P6: Rework due to the reappearance of already fixed bugs	
P7: Spaghetti branches strategy	Control the Time Lead
P8: A high rate of post-release defects	

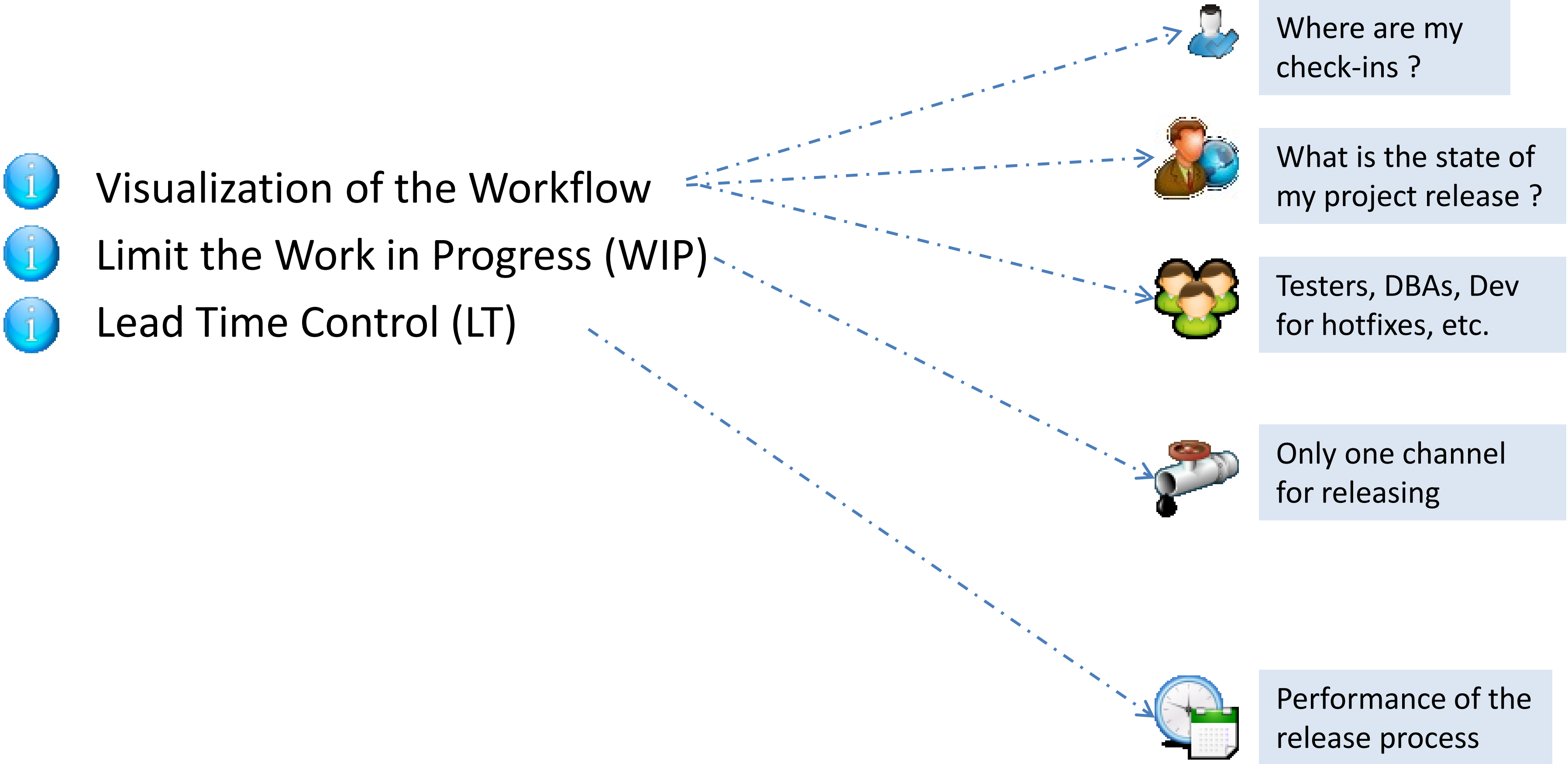
- Plus
  - Hurry to release this version !
  - Can we afford not to debug in the production environment !!
  - Improving collaboration and visibility. Thinking such as developers and sharing the production environment vision !!!

The release process (how it looks like?)



- ***Shared vision*** of the release process as modeled and verified by DSL4SPM tool
- Four ***Phases*** including different activities
- More than one ***role involved*** in the process of release
- Need of ***coordination*** between roles in each stage of the release process
- Highlights the ***checklists*** within each phase of the release process (DB scripts, conf, etc.)

KANBAN PRINCIPLES



- Visualization of the Workflow
- Limit the Work in Progress (WIP)
- Lead Time Control (LT)

Where are my check-ins ?

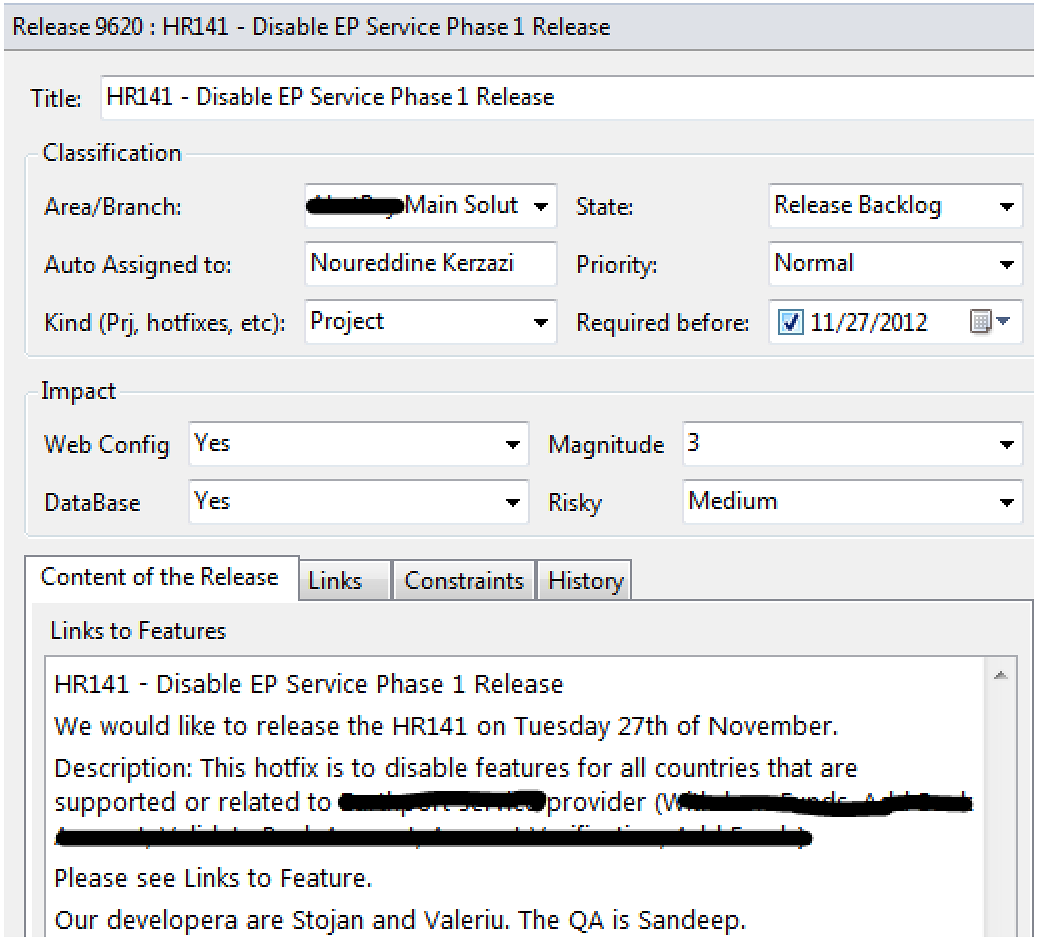
What is the state of my project release ?

Testers, DBAs, Dev for hotfixes, etc.

Only one channel for releasing

Performance of the release process

Release Work Item



Release 9620 : HR141 - Disable EP Service Phase 1 Release

Title: HR141 - Disable EP Service Phase 1 Release

Classification

Area/Branch:	[redacted]Main Solut	State:	Release Backlog
Auto Assigned to:	Noureddine Kerzazi	Priority:	Normal
Kind (Prj, hotfixes, etc):	Project	Required before:	☑ 11/27/2012

Impact

Web Config	Yes	Magnitude	3
DataBase	Yes	Risky	Medium

Links to Features

HR141 - Disable EP Service Phase 1 Release

We would like to release the HR141 on Tuesday 27th of November.

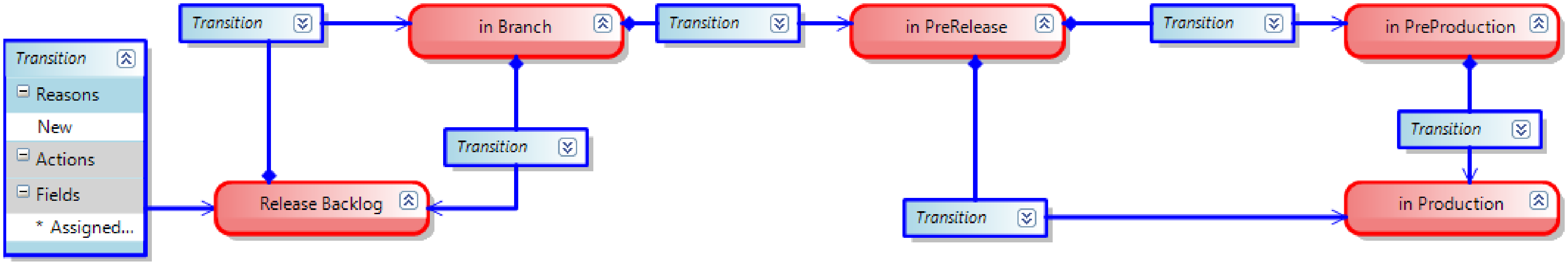
Description: This hotfix is to disable features for all countries that are supported or related to [redacted]provider ([redacted]

Please see Links to Feature.

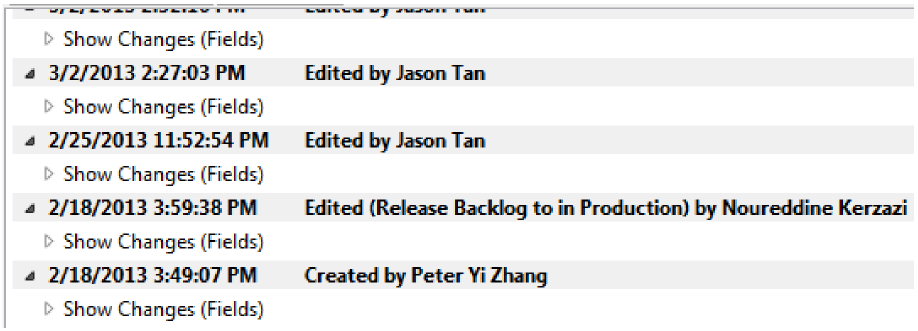
Our developera are Stojan and Valeriu. The QA is Sandeep.

- An adapted work item within the collaborative system (TFS)
- A manager requests the release of a given project by creating a new release work item.

Workflow of the Release Process

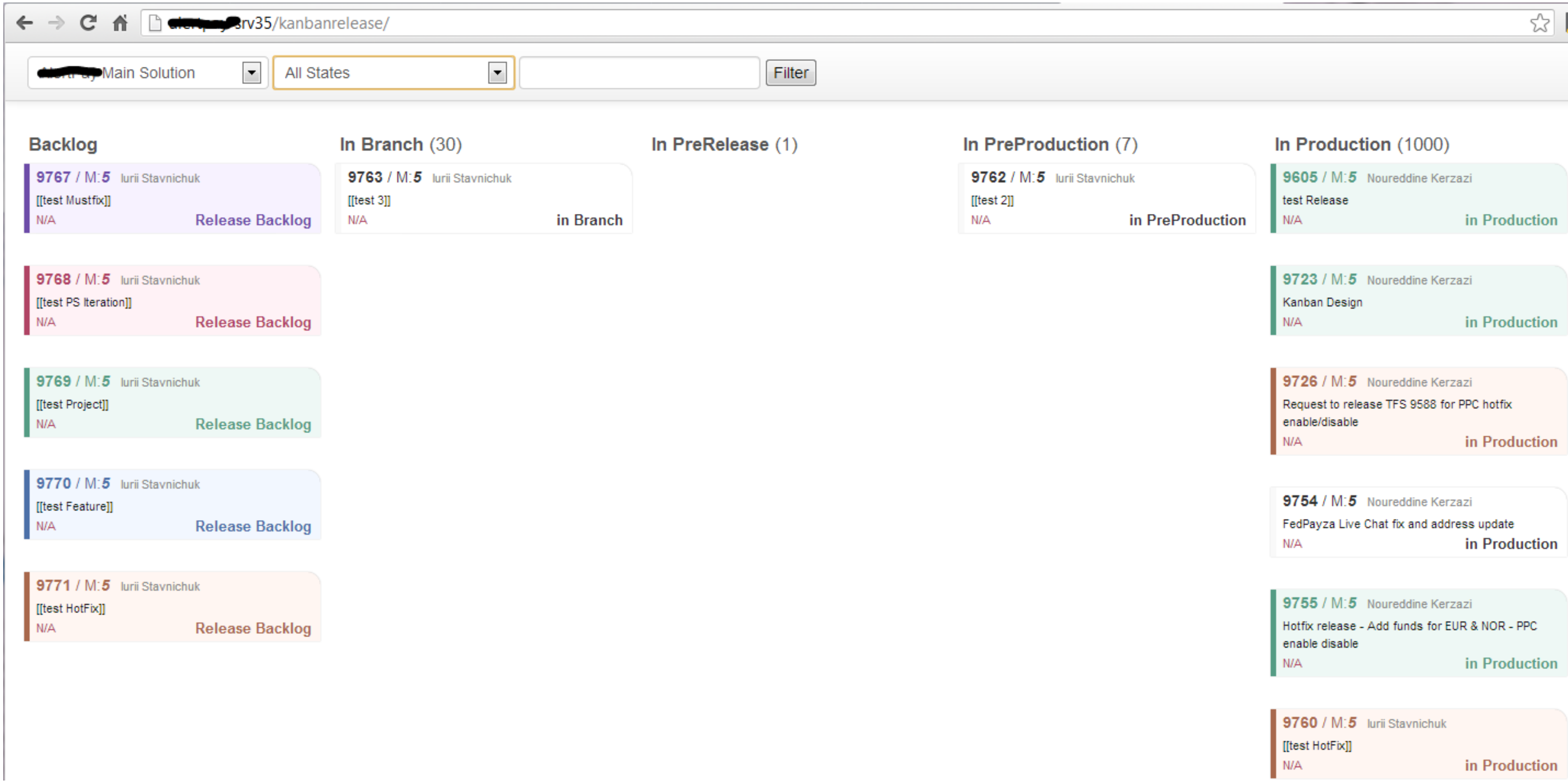


- Build on top of the internal collaborative tool (TFS)
- States in read (e.g., in branch)
- Transitions in Blue which include the Reasons, Actions, and Fields
- A history for each release work item



```
3/2/2013 2:32:16 PM      Edited by Jason Tan
  Show Changes (Fields)
3/2/2013 2:27:03 PM      Edited by Jason Tan
  Show Changes (Fields)
2/25/2013 11:52:54 PM    Edited by Jason Tan
  Show Changes (Fields)
2/18/2013 3:59:38 PM     Edited (Release Backlog to in Production) by Noureddine Kerzazi
  Show Changes (Fields)
2/18/2013 3:49:07 PM     Created by Peter Yi Zhang
  Show Changes (Fields)
```

Kanban board implemented as a website



[redacted]rv35/kanbanrelease/

[redacted]Main Solution | All States | Filter

Backlog

9767 / M:**5** Iurii Stavnichuk
[[test Mustfix]]
N/A Release Backlog

9768 / M:**5** Iurii Stavnichuk
[[test PS Iteration]]
N/A Release Backlog

9769 / M:**5** Iurii Stavnichuk
[[test Project]]
N/A Release Backlog

9770 / M:**5** Iurii Stavnichuk
[[test Feature]]
N/A Release Backlog

9771 / M:**5** Iurii Stavnichuk
[[test HotFix]]
N/A Release Backlog

In Branch (30)

9763 / M:**5** Iurii Stavnichuk
[[test 3]]
N/A in Branch

In PreRelease (1)

In PreProduction (7)

9762 / M:**5** Iurii Stavnichuk
[[test 2]]
N/A in PreProduction

In Production (1000)

9605 / M:**5** Noureddine Kerzazi
test Release
N/A in Production

9723 / M:**5** Noureddine Kerzazi
Kanban Design
N/A in Production

9726 / M:**5** Noureddine Kerzazi
Request to release TFS 9588 for PPC hotfix enable/disable
N/A in Production

9754 / M:**5** Noureddine Kerzazi
FedPayza Live Chat fix and address update
N/A in Production

9755 / M:**5** Noureddine Kerzazi
Hotfix release - Add funds for EUR & NOR - PPC enable disable
N/A in Production

9760 / M:**5** Iurii Stavnichuk
[[test HotFix]]
N/A in Production

- The content of Swimlines is based on the states of the Release work item

Post-Transition

- Evaluation on the Kanban Transition
  - Respect the culture of the organization (short time planning)
  - Makes the release process visible
  - More control on the Lead Time
  - Real test effort calculation along with accurate estimation of the release date

- Practical Implications
  - Identifying bottlenecks within the release process
  - Making the release planning more flexible
  - Moving from cadenced release cycle (1 each 2 months) towards a model of release on demand (n by week)
  - Notifying stakeholders about the progress of each release

Conclusions and Outlooks

- Conclusions
  - Useful approach when we are not able to have accurate plan of releases.
  - The executive dashboard provides visibility of each release progress.
  - Limit of WIP preserves the overheating of release team.
  - Control of Lead Time allows identifying the states that are time-consuming.
- Outlooks
  - Graphical system for monitoring the production environment after each release.
  - Debug obscure loader Errors tool such as Assembly Binding Log Viewer (e.g., FusLogViewer).
  - Simplify the deployment process (progressive release toward the farm of prod. servers).
  - Optimize automatic testing (focused tests based on dependencies).
  - Establish a new practice for root cause analysis of post-release defects.

Questions