Progress Manage

- Technical report -

Team:

Alexandru Rusu,

Daniel Popescu,

Daniel Unguru,

Claudiu Unguru,

Coordinators:

Dr. Ferucio-Laurențiu Țiplea,

Dr. Sabin-Corneliu Buraga

Contents

[Problem presentation 3](#_Toc473073627)

[State-of-the-art 3](#_Toc473073628)

[Solution 3](#_Toc473073629)

[Comparison with other solutions 4](#_Toc473073630)

[Future work 4](#_Toc473073631)

[Conclusions 4](#_Toc473073632)

[Bibliography 5](#_Toc473073633)

[Links 5](#_Toc473073634)

# Problem presentation

The existing solutions for managing workloads lack tools for teams monitoring and simultaneously comparison between several teams.

Existing solutions do not provide a single screen view over the metrics such as burndown chart, tasks blocked from current iteration, the hours spent over the initial estimate for all teams.

Another problem of the existing solutions for managing workloads is that the sprints can skip the required steps, for example you can set a task from the step “Estimation” to the step “Done” without passing through all required steps: Estimation – Implementation – Testing – Done.

# State-of-the-art

The existing solutions implement application lifecycle management (ALM) methodology.In summary, ALM is a set of predefined processes that start somewhere in the business as an idea, a need, a challenge or a risk and then pass through different development phases such as requirements definition, design, development, testing, deployment, release and maintenance spanning across an entire lifecycle of a product. Throughout the ALM process, each of these steps is closely monitored and controlled, followed by proper tracking and documentation of any changes to the application.

The existing solutions are very useful for tasks management but don’t offer a good support for comparison between teams.

# Solution

The aim of the project is to create an application that will check the validity of the workflow graph and will provide a centralized view to compare the progress of the teams within an IT company taking into account their working methodology (kanban, waterfall, etc.), tools used, number of members, their capacity of work and other metrics.

The Web application will provide a simultaneously comparison between several teams in order to improve the future work - the teams will learn from each other.

The implementation will adopt a service oriented architecture (SOA) style - important functions can be accessed via a REST API.

The application will be written in a reactive style and will take advantage of four big benefits of design (responsive, resilient, scalable and message-driven).

The proposed app will collect only metrics about tasks management and will skip the release and maintenance part, because those metrics are not relevant in our context making the app useful not only for the team working in IT but for all teams that working with online task management services across the industries.

Implementation of the app will extend workflow management systems with planning facilities also will use Petri nets approach that successfully accomplishes this task since Petri nets are well suited for capturing flows in web services, for modeling the distributed nature of web services, for representing methods in a Web service and for reasoning about the correctness of the flows.

# Comparison with other solutions

Some relevant of ALMs examples would be:

1. **Jira** isa proprietary issue tracking product, developed by Atlassian. It provides bug tracking, issue tracking, and project management functions.
2. **TFS**. Team Foundation Server is a Microsoft product that provides source code management, reporting, requirements management, project management (for both agile software development and waterfall teams), automated builds, lab management, testing and release management capabilities.
3. **Sagitta-2000**. Started in 1994 by the Dutch Customs Department as a nationwide information system for handling customs declarations.

Primary goal of this project is to define the core customs declaration processes and implement them in workflow software.

1. **AIMES**. Petri net-based process monitoring: a workflow management system for process modelling and monitoring. The goal is to improve medical equipment management.

# Future work

Future work will involve the creation of new tools for comparison such as a visualization between the roles (developer, business analyst, tester, etc.) of the team members for a better understanding of the time consumption and resource allocation.

# Conclusions

The proposed solution check the validity of the workflow graph and will provide a centralized view to compare the progress of the teams.

# Bibliography

1. Nick Russell Arthur H. M. ter Hofstede - Formal Approaches to Business Processes through Petri Nets
2. Sergi Mansilla - Reactive Programming with RxJS Untangle Your Asynchronous JavaScript Code
3. W.M.P. van der Aalst. Putting Petri nets to work in industry. Computers in Industry, 25(1):45–54, 1994.
4. W.M.P. van der Aalst. A class of Petri net for modeling and analyzing business processes. Computing Science Reports 95/26, Eindhoven University of Technology, Eindhoven, 1995.
5. W.M.P. van der Aalst. Three Good reasons for Using a Petri-net-based Workflow Management System. In S. Navathe and T. Wakayama, editors, Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC’96), pages 179– 201, Cambridge, Massachusetts, Nov 1996.

# Links

1. https://gist.github.com/staltz/868e7e9bc2a7b8c1f754
2. http://martinfowler.com/articles/serverless.html
3. https://d3js.org/
4. <https://en.wikipedia.org/wiki/Application_lifecycle_management>
5. <https://www.visualstudio.com/vsts-test/>
6. <https://www.atlassian.com/software/jira>
7. <http://wwwis.win.tue.nl/~wvdaalst/publications/p53.pdf>
8. <https://gist.github.com/staltz/868e7e9bc2a7b8c1f754>
9. <http://martinfowler.com/articles/serverless.html>
10. <https://en.wikipedia.org/wiki/Jira_(software)>
11. <https://en.wikipedia.org/wiki/Team_Foundation_Server>
12. <http://link.springer.com/article/10.1007/s10009-007-0038-x>
13. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.33.960&rep=rep1&type=pdf>
14. [Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications](https://books.google.ro/books?id=PNmoCAAAQBAJ&pg=PA537&lpg=PA537&dq=Sagitta-2000&source=bl&ots=NCPL3TMzix&sig=11EgtyPjT2z3fpVOxMLq31rWVP4&hl=en&sa=X&ved=0ahUKEwje3Yu1qODPAhXCiCwKHfFOAsUQ6AEIPzAK#v=onepage&q=Sagitta-2000&f=false)
15. <https://medium.com/reactive-programming/what-is-reactive-programming-bc9fa7f4a7fc#.u1j2fvkzj>
16. <http://www.dis.uniroma1.it/~marrella/slides/Sem_PM_13-14_BPs_Fundamentals.pdf>
17. <http://link.springer.com/article/10.1007/s10845-012-0704-z>