

**Can the way that user stories are prioritised  
and allocated I.E. "MoSCoW method"  
have an impact on the length of a sprint in  
the video games industry?/ Are User Story  
distribution methods used in regular Agile  
Development practice useful in the Games  
Industry?**

**COMP150 - Agile Development Practice**

1606119

November 22, 2016

My agile development question revolves around one specific part of the Agile Development process and that is "User Stories", and how I intend to answer the aforementioned question is to delve deeper into first, what a User Story is and how it is used by members of the development team in the development cycle to create a finished product, and then go further and look at how the user stories are created in the first place, and thus prioritised

amongst the team members and on the Scrum Board, and as there are different methods of prioritising these user stories such as Walking Skeleton and the MoSCow method, I will compare and contrast these individual methods, and by using my research gained from my noted sources, I will be able to ascertain which is the most efficient for teams in a games development environment, and of the list are going to impact development time the least. And from these reflections on the different methods and the ways that they change the development teams behaviour and work pattern I will be able to draw a final conclusion on my question and whether further research is required to gain a better understanding of the impact it can have on the development time of a project in the video games industry.

## **1 Introduction**

As the principles and philosophy of Agile Development is now common practice in both software and games development, the smaller parts of the overall "Agile" umbrella become more and more important to pick apart and evaluate to make sure that development goes as smoothly as possible and that "valuable software" and software is created "early and continuous" [1], so that projects are delivered to customers on time, which after all is the first point of the Agile Manifesto. And as User Stories are the way that work is presented to members, their creation and distribution must be closely monitored to ensure that the scope of the project is not under or over evaluated, and that members of the team are not over encumbered with work or tasks they can't carry out efficiently, which would extend the length of a sprint; and there are several methods of User Story distribution that I have discovered through my research that are generally used in Agile Development, and my research has found that the majority of them do help but aren't carried out

in video games due to the different development practices and outside factors overriding their usefulness, but the main takeaway from my essay is how User Stories could be better distributed to aid the security of the length of sprints.

## **2 Research Findings**

Through my research, I have found that there is quite a divide on my topic, and the usefulness of such distribution methods, with my first source outlining a relatively large method that according to the source is "really simple and easy to understand" [1], which is always important when working under such tight time constraints and milestones that are found in the modern Games Industry, however later they state that the method is only suitable for the "prioritization of small and medium size project can be calculated efficiently" [1], which makes it good for general software development and Indie Games, where a usually multi-skilled team is working on a project and thus can cover several disciplines, however it seems to me that it would be quite redundant for typical AAA Development as staff are normally very specialised and won't normally cross over to other disciplines in normal development practice. To further back this up, Popli and Chauhan and Sharma even goes as to say that "Prioritization of user-stories is a difficult task in agile environment because of its volatile nature." [4] which is especially true of games development as there are many factors that can affect the length of a sprint, as mentioned by Al-azawi, Ayesh and Obaidy, such as "Feature Creep" [3] which entails "new functionality is added during the development phase to increase project scope and change schedule time" [3] and these type of events are what I would describe as "Volatile" and such can affect the sprint length on a project, along with causing other issues such as "increased error, possible defects and increased chances of failure" [3] which again in turn

can also lead to increased sprint time.

### **3 Conclusion**

Write your conclusion here. The conclusion should do more than summarise the essay, making clear the contribution of the work and highlighting key points, limitations, and outstanding questions. It should not introduce any new content or information.

### **References**

- [1] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas, “Principles behind the agile manifesto,” [Online]. Available: <http://agilemanifesto.org/>, [Accessed: 21-Nov-2016].