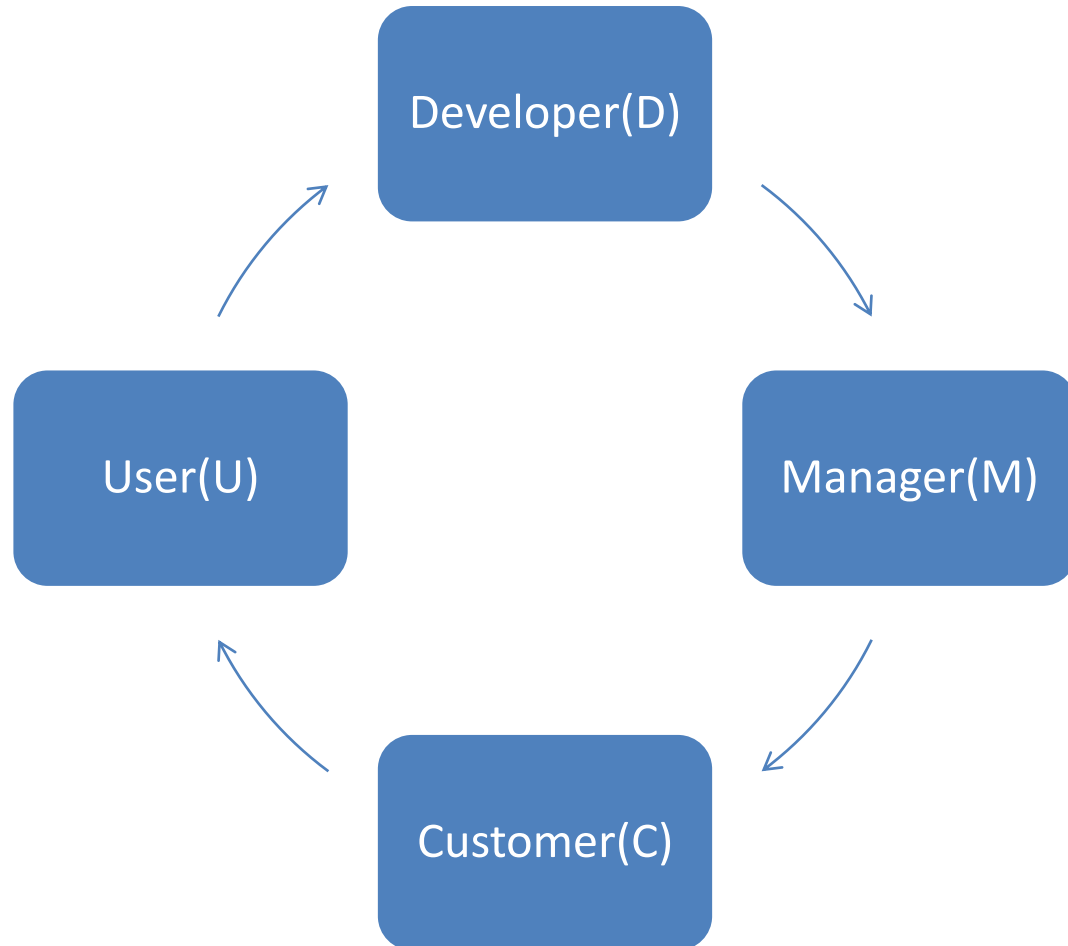


Effectiveness / Time

The good side...

The Stakeholders



Focus on the Customer

Easy to change requirements

- reduced need of formality(D), Less rework(D), Changes can be added to current iteration(D, M, C)

On-site Customer

- => fewer misunderstandings and therefore less rework(D,C)

Release Regularly

- => Shorter time to market(M,C,U)

Focus on the Developers

Developers empowered

- => higher motivation => increased productivity(D,M,C);

Collective code ownership

- => code belongs to the team, and any developer can make minor changes without a need to consult the entire team(D).

Continuous integration

- => master branch always up to date with the latest feature implemented: fewer conflicts in the end, they are solved as they come up.. reduces the complexity of merging different features.(D,M,C)

Pair-Programming

- => Shared knowledge, improvement on the code quality (on long run fewer bugs)(D,M)

Timeboxes:

- set dates for start and end of collection of activities => keeps developers focused

A few numbers

- “A 10x reduction in inventory in the system. A 5x reduction in WIP. A 3.4x increase in productivity with no new money, resources, people or any change in the way software engineering (development and test) were conducted.”
 - http://www.agilemanagement.net/index.php/blog/HP_gets_3.4x_productivity_gain_from_Agile_Management_techniques/

A few numbers (cont.)

- “The team’s productivity has been compared with industry data via the Galorath database, QSM’s database, and Capers Jones’ database, and in every case they come out among the top performers. From the data, no one could distinguish them from the best teams in the software industry, yet they were missing a substantial degree of qualifications for this work. That gap was overcome by agile software development techniques and the presence of senior level skills among some team members.”
 - <http://www.vantage-interactive.com/BigBand/file.php/1/Agile%2520by%2520the%2520Numbers-NancyV.pdf>