

The need for a Common Lingua Franca in Agile Working environments

Anand Sunder
Capgemini

Abstract: Touching the annals of extreme programming we reach a point where the principles of agile manifesto must work in cohesion to evolve a common mode of communication. We discover this increasingly as environments promoting software development reaches their saturation levels. The effort involved in translating business specifications to code is exponentially rising, this needs to be watched over prudently by participants of the development lifecycle.

Here we delve deeper into the fourth principle of agile, wherein businesspeople and developers must work together throughout the project duration (Scrum Alliance 2021,e.t al).

Upon closer investigation we realize that there are 4 decision points:

- i) Businesspeople
- ii) Developers
- iii) Product approval
- iv) Development issues

The deeper aspects of synchronization involve trying to bridge the gaps in not only the timelines of delivery expected, but also the mode of communication between various participating entities trying to bring the project to fruition.

Critical challenges include the gaps in meeting business expectation and technological outcomes of software delivery. This is talked of in detail subsequently in the paper.

Keywords: Agile, Common Lingua franca, Business Expectations, Technological Outcomes, Extreme Programming

INTRODUCTION:

A closer investigation into the business lifecycle tells us that there is continuous interaction between different participants in an agile working environment. Scrum Master, Product owner, business analysts, testing team and developers carry their own unique experience set and this is brought to fruition in the form of incremental product churn outs. Each skill has a distinct rate and methodology of working, which forces rest of the participants to mend their style of working around it. Quite in contrast to the waterfall methodology of working, where there is a prerequisite to each subsequent step agile brings out emphasis on collaboration and team work to a greater extent.

In the waterfall approach (Sarah Lewis, e.t al) the seven non-overlapping stages include:

- i) Requirements: R
- ii) Analysis: A
- iii) Design: D
- iv) Coding/Implementation: C
- v) Testing: T
- vi) Operation/Deployment: O
- vii) Maintenance: M

While one stage is the prerequisite to the next with no feedback in sight, there is no need for a communication mechanism that is common to all, everyone knows their share of the task and has a mechanism evolved to translate it into the format they are expected to deliver or produce. Based on no feedback, sequential flow we come up with the following mathematical functions to model each of these:

$$A = f(R, t) \dots \dots \dots (1)$$

$$D = g(R, A, t) \dots \dots \dots (2)$$

$$C = h(R, A, D, t) \dots \dots \dots (3)$$

$$T = i(R, A, D, C, t) \dots \dots \dots (4)$$

$$O = j(R, A, D, C, T, t) \dots \dots \dots (5)$$

$$M = k(R, A, D, C, T, O, t) \dots \dots \dots (6)$$

$\forall i \% j \neq 0 \text{ S.T } i, j \in \{A, D, C, T, O, M\}$ (Constraint for zero feedback)

For agile flows equations (1) through (6) hold good except that the constraint changes to be aiding a feedback-based system, i.e

$\forall i \% j = 0 \text{ S.T } i, j \in \{A, D, C, T, O, M\}$ (Constraint for feedback in mechanisms)

We delve deeper into a very special case of agile, focusing specifically on the fourth principle where businesspeople must work with developers throughout the business duration.

In the flow chart below, we see that businesspeople give requirements to developers, who translate it to a product, the product gets validated before which fresh set of requirements are given back to developers.

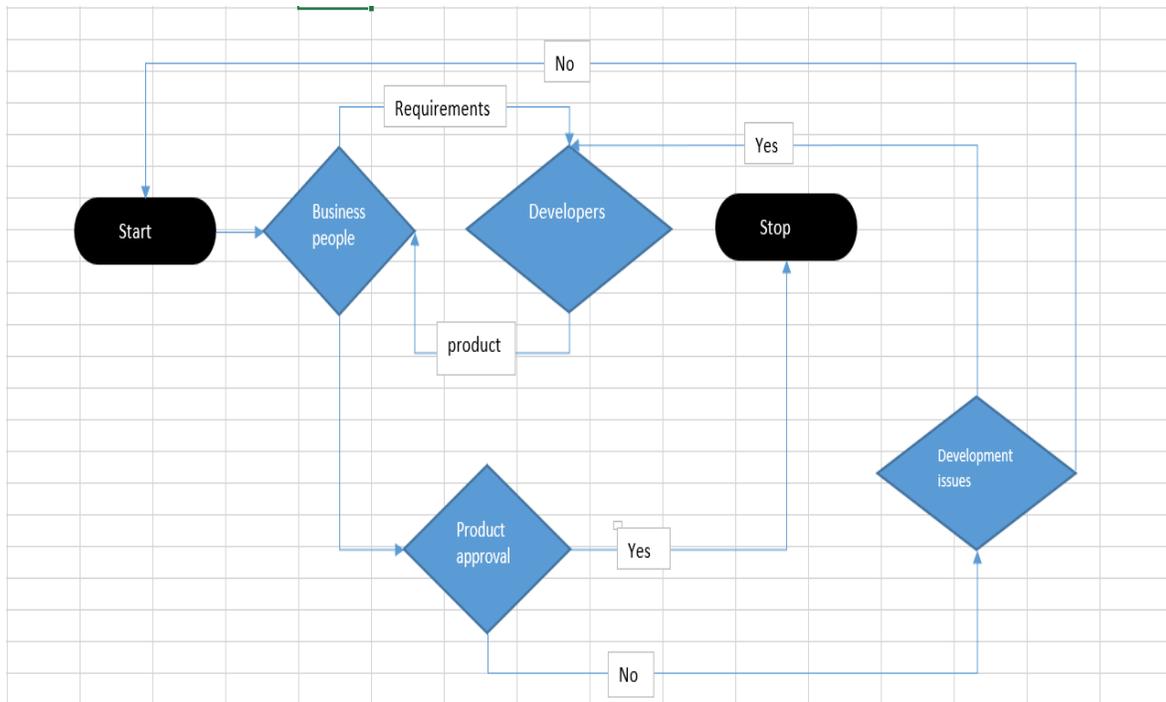


Figure: The fourth principle of the agile manifesto illustrated above.

When businesspeople do not approve of the product it could be for issues which are:

- i) Development related
- ii) Not development related

From the above schematic we have:

$$R = h(BP)$$

$$P = f(R) * DI$$

$$\forall DI =$$

1 for no Development issues and 0 otherwise (only if Product Approval (PA=No or 0))

$$PA = g(BP) \text{ (BP is Business people)}$$

CONCLUSION:

Between businesspeople and developers thus a mode of communication must be evolved which is common to

them and bridges the divide, facilitating unhindered communication the additional decision blocks shows would eventually disappear.

REFERENCES:

- 1) The Agile Manifesto :https://www.scrumalliance.org/resources/agile-manifesto?utm_term=&utm_campaign=Live+Online+Courses&utm_source=adwords&utm_medium=ppc&hsa_acc=8637613312&hsa_cam=10054990704&hsa_grp=112186221774&hsa_ad=498878865721&hsa_src=g&hsa_tgt=dsa-1025284834960&hsa_kw=&hsa_mt=b&hsa_net=adwords&hsa_ver=3&gclid=EAIaIQobChMI1eOc3-SR8AIVLpZLBR0ScwLaEAAYASAAEgLZPPD_BwE
- 2) Waterfall Model : <https://searchsoftwarequality.techtarget.com/definition/waterfall-model>