Table of Contents

Useful References

Idea here is to maintain a list of references I find useful in helping my understanding of all things Agile.

- Teams with Remote People
- Canonical References on Scaling

Videos Useful in Training

Note: to download when you cannot get to YouTube online, put "ss" in front of YouTube in the address.

When training, some of these videos can lighten things up:

- New Zealand All Blacks Haka (this is the meaning of Scrum? no ...)
- Jeff Sutherland's TedX video on doing "Twice the Work in Half the Time"
- VersionOne's description of the Agile Manifesto
- Drive: The Surprising Truth About Motivation plus Dan Pink's TED Talk
- Scrum in 7 minutes
- The Rong Way to Do Agile: Team Structure from Atlassian
- Writing Good User Stories from Rally
- Product Backlog Example to build web site for Scrum Alliance (from Mike Cohn)
- Planning Poker for Estimates from Mike Cohn
- Example Ball Point Game
- Prudential Ad Showing That We Are All Optimists When Considering the Future
- Why Cost of Delay Matters?
- Sinek's Ted Talk - Start with Why - Bit from 1:35 to 5:15 relevant for product owners, for example, when explaining "vision".
- High Performance Tree - Lisa Adkins on High Performance teams
- How The Brain Stores Information - TED Talk on importance of visual processing etc.

Also videos of games:

- Airplane game
- Ball Point Game

Other Useful Links for Training

- 12 Principles of Agile Development by Julien Henzelin
- VersionOne "State of Agile" Survey (9th Edition). You need to register to get it - my version.
- Help answer question "What does a Scrum Master do?"
- Base approach to splitting stories. Additional thinking at from Bill Wake 20 ways to split stories. These ideas are to help to get to a minimal, end-to-end solution present which usually has high

value, from which you can thin then fill in the rest of the solution in subsequent iterations.

- [Role of Product Management from Pragmatic Marketing framework](#)
- [The Burning Platform metaphor](#) by Daryl Conner. Idea is to have the resolve to make the change happen. Resolve can come from current or anticipated problems or current of anticipated opportunities, not necessarily as a result of "peril" (which is "current problem" classification.
- [Alistair Cockburn etc on Communication](#) including richness chart and the implications of this richness.
- [James Coplien on the Borland Quattro Pro development](#) which some say was the most productive ever recorded. To quote the abstract "The project assimilated requirements, completed design and implementation of 1 million lines of code, and completed testing in 31 months. Coding was done by no more than eight people at a time, which means that individual coding productivity was higher than 1000 lines of code per staff-week. The project capitalized on its small size by centering development activities around daily meetings where architecture, design, and interface issues were socialized. Quality assurance and project management roles were central to the development sociology, in contrast to the developer-centric software production most often observed in our studies of AT&T telecommunications software. Analyses of the development process are "off the charts" relative to most other processes we have studied."
- [Discussion on roles beyond PO, SM, and team basics](#)
- [Advice on Running Scrum-of-Scrums meeting](#) from Mike Cohn.

Sources of Ideas

- [Tasty Cupcakes](#) a source of interesting games you can run to get a message across.

From:
https://www.hanssamios.com/dokuwiki/ - **Hans Samios' Personal Lean-Agile Knowledge Base**

Permanent link:
https://www.hanssamios.com/dokuwiki/useful_references?rev=1445707122

Last update: **2020/06/02 14:28**