Table of Contents

Useful Videos 3

Useful Videos

A list of videos I've come across over the years that help get concepts across.

- Don Reinertsen - Second Generation Lean Product Development Flow. Talks about some of the content of his book. As usual a very “dense” pitch with a lot of information. If you want to understand impacts of things like queuing theory, lean, and why variability should be preserved for new product development (hint: Black Scholes option pricing model) then this is the video for you. See My notes for more information.
- Dan Pink’s TED Talk. Or if you want the more fun presentation then use Drive: The Surprising Truth About Motivation. Helps you understand how knowledge workers are motivated: autonomy, mastery, and purpose (and take money off the table)
- Dave Thomas - Agile is Dead. Provocative title which talks about how implementations of agile don’t always match up with the specification of agile.
- Martin Fowler - Microservices. Found this to be a useful discussion to help me talk to technical people.
- “Uncle” Bob Martin - The Principles of Clean Architecture. Great anecdotal story at the beginning, as always. Then discussion around how architecture should reflect the domain - what the application does.
- 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
- Scrum in 7 minutes
- The Rong Way to Do Agile: Team Structure from Atlassian
- 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 - To understand how to communicate with people especially as you introduce something new. 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.
- Kenny Rubin “Essential Scrum” on Requirements and Change Management
- Kenny Rubin “Essential Scrum” on Product Backlog Refinement
- Jeff Sutherland on the Daily Scrum
- S&*% Bad Scrum Masters Say - Funny
- Henrik Kniberg on the Product Owner role - Key idea “Product Owner must say ‘no’”.
- Lyssa Atkins on Scrum in about 10 mins - Every Scrum Master should know how to explain the framework.
- Dave Allen - Teach Kids About Telling the Time - Funny video to understand how slippery the english language is to drive requirements.
- Steven Johnson - Where Good Ideas Come From - On understanding how innovation works - requires a collision of half ideas (the slow hunch) that have been fermenting in the background for a while. So idea is to provide an environment to connect. “Chance (of innovation) favors the

connected mind."

- The Backwards Bicycle - Great video to understand how its hard to unlearn what we know, that knowledge isn't the same as understanding, to learn you have to practice, practice, practice, and that you have biases and are unaware of them. See more at Why Is Agile So Hard - The Backward Bicycle?
- Leeroy Jenkins - What happens when 1 person doesn't consider the rest of the team. From World of Warcraft.
- Day in a Life of Mob Programming. Helpful to talk about learning and trying practices even if we don't adopt wholesale.
- Dilbert has "the knack". Funny video about becoming an engineer.
- Rugby game. Shows structure emerging from chaos, minimal control, common goals, etc
- Introduction to DevOps
- Agile Manifesto Explained - Quick 3 minute video on the basics of the Agile Manifesto, including a bit of history.
- Are we in Control of Our Decisions: Behavioral economist Dan Ariely, the author of Predictably Irrational, uses classic visual illusions and his own counterintuitive (and sometimes shocking) research findings to show how we're not as rational as we think when we make decisions.
- Coordination Chaos: Good video to help explain why the old way of working no longer works as the organization grows.
- High-tech Anthropology at Menlo: Video to explain how gemba helps when working on understanding customer requirements.
- Submarine Leadership: David Marquet on changing the leadership model from "leader - follower" to "leader - leader".
- 5 Dysfunctions of a Team: Patrick Lencioni presenting the materials of the book
- Wisdom of the Crowds demonstration counting gum balls: Useful to help people understand how even uninformed people can contribute to a discussion in estimation.
- To help people understand small vs large batch processing (the ideal of one piece flow in manufacturing world) when you cannot run something like the penny game:
    - Batch of 10 vs batch of 1, simultaneously
    - Batch of 10 vs batch of 1, serially
- Dave Snowden on Organizing a Children's Birthday Party
- The Power of Empathy - For Leadership
- Locating Yourself - The Key to Conscious Leadership - Are you operating above the line or below the line.
- Design Thinking - Introduction to the basic ideas. While it is presented as a "linear" process, and misses notions of divergent and convergent thinking, it is a good start.
- How to Use the Customer Empathy Map
- Eric Ries on Innovation Accounting - How do we know we are making progress when all we are doing experiments.
- The Lucky Iron Fish - Helps people understand why we need to do "gemba" (go and see) when trying to understand the requirements of the product.
    - TEDTalk on How the Lucky Fish Can Treat Anemia - Useful video that expands on the discussion above to look at a complete process of making a product "fit for purpose" including the design of the product, the implementation of the product, and the delivery of the product.
- How to Trust People You Don't Like - Podcast dispels a number of misconceptions about trust.
- How to Write Mission Statements That Don't Suck - stop over wordsmithing your mission statement and avoid "solutions"

- 10 Reasons Estimation and Planning Fails and What to Do About It by Troy Magennis. Great pitch. Love the basics here. Main message is “a lot of bad things happen in projects which increase utilization into the 'non-linear' (when lead-time graphed over utilization - above 80% utilization leads to exponential lead time) zone and so make estimating useless and forecasting difficult.” Key learnings include:
  - If you are in the non-linear region of utilization for your project (i.e. greater than 80% utilization) then estimation will always fail.
  - Sometimes you need just enough information to decide what not to do, and perhaps toss a coin for remaining options (if they really are close enough).
  - If we only have a high utilization system and we cannot change this, then estimating using points etc doesn't make sense. For high utilization systems we need to track system level impediments to the flow.
  - Top 10 list (portfolio level): (BTW why this project won't finish on time is because of last project - and no amount of estimating is going to help)
    - Don't start on time. Eg delay in previous project
    - No team (no one to do the work).
    - Partial team. Eg on prior project. Half strength team → over utilization
    - Partial body substitution. Similar to above.
    - Missing skills sets. Variation on a theme. Are we producing useful stuff. Look busy but don't know whether we are making progress. See “capability matrix” based on level of capability - novice and learner (but will to learn), do / maintain and so able to modify (and make small additions), teacher / creator (able to do work and show others). Help actively develop your T-shaped people.
    - Over-stated parallel effectiveness. Adding teams / people to make it faster does not scale linearly. Limited by whatever serial processes you have in place. Variation on Amdahl's but applied to people. Spend 8x the effort but only 3x additional capability. Need to invest in independents and tools (eg continuous deployment)
    - Dependency and friction. Tracing through how work was actually done. How many levels of dependency. 1.5X the Sprint length is the fastest you could move through the system (as sometimes it doesn't get delivered in the Sprint expected.) Need to understand the dependencies you have. Every dependency you can remove from your delivery stream doodles your chances of delivering on time. This could mean, for example, having merged larger teams to reduce the number of dependencies.
    - Carried over defects and debt.
    - Ship stoppers. There are always these. If you ask the question “if I double the team would this help”. Often answer is “no”. Tried to find these out early eg by surveys every week “would you ship this tomorrow …”
    - Splitting. Rate the we finish items and the rate that we arrive are never the same, because we split. We need to take into account that we are splitting items as we do work.
  - Use these “assumptions” to understand whether the project is on track. Don't really need status reports if these assumptions are not met.

Videos, Learning, Improvement, Consultant

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

Permanent link:
https://www.hanssamios.com/dokuwiki/useful_videos?rev=1645808494

Last update: **2022/02/25 09:01**