

Real Life Examples & Anecdotes from Lean Agile Software Development Coach

Lean and Agile Software Development

- Aka LASD
- An approach to software creation / solving business needs
- Focuses on team collaboration to gain shared understanding
- Visual (group) modelling using lightweight, low fidelity techniques with UML – user centred thinking
- Outside-in engineering
- Hands-on, collaborative – including TDD and ATDD
- Courses created by Craig Larman – introduced at JPMC 2013
- Delivered by in-house teachers and used for some years on the graduate programme

Agility and Leadership: Thoughts and Reflections



“What if we don’t change at all ...
and something magical just happens?”

- Effort
- Energy
- Patience
- Grit

Change in technology

Why do technology teams sometimes struggle?

It's not because they lack skill [necessarily], it's because

- Change is ambiguous
- Change is constant

Implementing agility, must include the ability to increase **learning velocity** - it's not just about delivering quickly

Furthermore boundaries, responsibilities and authority levels are often poorly defined when adopting agile

Why do [Agile] transformations fail? (or are at least sub-optimal)

Not because of tooling

It's because of leadership behaviours (on the whole)

- It's not a process implementation – it's a mindset shift
- Measuring output rather than outcomes
- So called self-organising teams being "managed" by command and control style

Agile exposes leadership gaps

Leadership must mean enablement

Encourage "leader-leader" models

- Set vision and direction
- Remove impediments and constraints
- Allow time and safety for innovation/experimentation
- Build capabilities
- Shift governance to teams

What is required of "agile" leaders...

- Provide psychological safety
- Adaptive stability in teams
- Balanced autonomy and alignment
- Technical excellence – one of my favourites!
- Decentralised (where appropriate) decision making
- The ability to deal with resistance to change
- Provide clear intent with guard rails

Agile should be a leadership philosophy – not a project to implement

Some coaching/leadership stories

The tale of the:

- command and control leader
- disruptive, ego-driven developers
- CEO and Chief architect who didn't understand teams
- best working fun I ever had

My key takeaways (what I focus on)

- Engineering discipline to win
 - Automate, automate, automate
 - Clean code
 - Effective documentation (a LASD lesson)
- Refinement – demand this of your teams
- You need allies and competency
- Compromise and pragmatism are important... but sometimes!

Final thoughts

- TDD is still awesome and even faster with AI
- Be a teacher
- Don't fear change – it's coming regardless