Lean vs Agile vs Design Thinking

- **All 3 disciplines work on short cycles.**
- They should all be working together on common problem but truth is none of these groups work in tandem
- Need a clear understanding of the problem
- Teams that want to build cross team collaboration don't know where to start – should we build a design studio, define an MVP, when do we ship code, how do we measure success, who leads?
- Teams naturally pull away from each other rather than working toward a common set of objectives.

1. Tech teams are learning Agile and are focused on releasing bug free code through Sprints SAF – release trains) and focused on velocity and efficiency vs value

    - Limit feedback to the stakeholder not the end user
    - Without a clear understanding of the customer engineers shipped features, the more the better without a sense of helping a customer need in an effective way
    - Waterfall is old school, requirements are incomplete or inaccurate, takes too long, market can shift before you release, uncertain about customer learnings
    - DevOps – ship code in a continuous state and there vs version launches, net usage yields feedback and offers agile approach to change
    - Downfall is how to implement in scale i.e., roll out to dozens or hundreds and the gains are gone
    - SAF – scalable agile framework
    - **Can we build it.**

2. Product teams are learning Lean are focused on driving efficiency, quality and reducing waste through backlog tactical prioritization (emerged from LEAN Thinking but has nothing to do with it)

    - Ends up with MVP, a bastardized term, and used in place of Phase 1 i.e., what are the least number of features we can get away with and still ship this product. **Teams are forgetting to learn before moving onto the next release.**
    - Product managers prioritize backlogs of work based on gut instinct and subjective input from stakeholders.
    - Everything we build is an experiment to gain feedback. Next build only happens if there are positive learnings from customer behaviours (did customers like the new capabilities, are they using the new capabilities, will they pay for the new capabilities).
        - What's the most important thing we need to learn on our project? This all about risk ie., what's the thing most likely to make you fail?
        - What's the least amount of work we can do to learn that? The real meaning of an MVP. Not about being lazy, but a commitment to do less work before committing to doing a full build out of the feature.
    - If there aren't positive learnings you PIVOT.
    - The Lean Start Up has problems making its way into company's cultures and delivery cycles. Main challenge is that it introduces uncertainty into the process. Specifically, if we're committed to a date and a roadmap but find out we're headed in the wrong direction what do we do now?! If we can't make these adjustments at the same pace we're learning them are we truly agile?

- Innovation lab – doesn't have a P&L. Has a design space with lots of beanbag chairs but rarely a strategic mandate and a clear path for transitioning a product idea (once validated) into an integrated production track with a broader organization. Production team doesn't have the same enthusiasm for the ideas hatched by the innovation group. Plus their so backed up with current commitments the new ideas get pushed.
- **Should we build it. And if the answer is yes, can we build a sustainable business model around it.**

3. Design teams are learning Design Thinking focused on bringing the customer problem front and center by validating problem/solution fit with DT activities, but their process is seen as long and drawn out and delayed product launch.

- Take an empathetic look at the products you're building for customers to understand the core needs being addressed. Then through a set of brainstorming sessions come up with a set of solutions to meet those needs in a way that is technically feasible and viable for the business.
- Design sprints to increase customer empathy.
- Without feasibility or strategic alignment of their designs, suggestions never stood a chance of seeing the light of day.
- Ideas too complex to implement, fail to include broader spectrum of collaborators.
- Risk adverse cultures that focus more on bonuses rather than the customer experience never stood a chance.
- **Are we solving a real problem, for a real customer in a meaningful way? How do we deliver something of value?**

Recommendations:

- Work in small cycles
- Run an experiment
  1. Get all 3 teams out in the field working together through observation
  2. Hold a retrospective after the Sprint
  3. Put the customer at the centre – if they're not using the product, they're not the customer. How do we know our customers value what we're shipping?
  4. Go ask what's working
  5. Balance work of MVP with only highest risk assumptions.
  6. Do less research more often. Use a cross functional team and test 3 customers not 12.
  7. Work and train as one balanced team. Staff the project with designers, engineers, product managers…under one team. Members should be independent and empowered to make decisions.
  8. Radical transparency. Be clear on why you're trying something new. Be clear on what success looks like and how you'll measure them. Post those metrics and progress against goals.
  9. Review incentive structures and criteria. Incentivize on collaboration and continuous learning. For example, you are rewards if the customer finds value in the features of what you have shipped. Teams will figure out the right balance between the 3 methodologies and structure their team accordingly.
  10. Make product discovery work a key component of your backlog. The work that gets visualized gets done. Not just project plan but add a learning plan.