Combining Agile with Design Thinking for Product Development

DHAWAL BHATIA

HENA FATMA OSAMA

Index

Abstract

Why would a work process that has traditionally been used by designers be relevant in the corporate or engineering world?

It’s simple and logical: designers systematically extract, teach, learn, and apply human-centric techniques to solve problems in a creative and innovative way. So, Design Thinking is inherently optimistic, constructive, and experimental. In theory, it is a great vehicle to be more innovative, to better differentiate your brand, and to shorten the time-to-market any product or service.

Design thinking on its own is not the bringer of change but when it’s combined to Agile Software Development Lifecycle can do wonders for a product organization, it is sure to bring a winning formula.

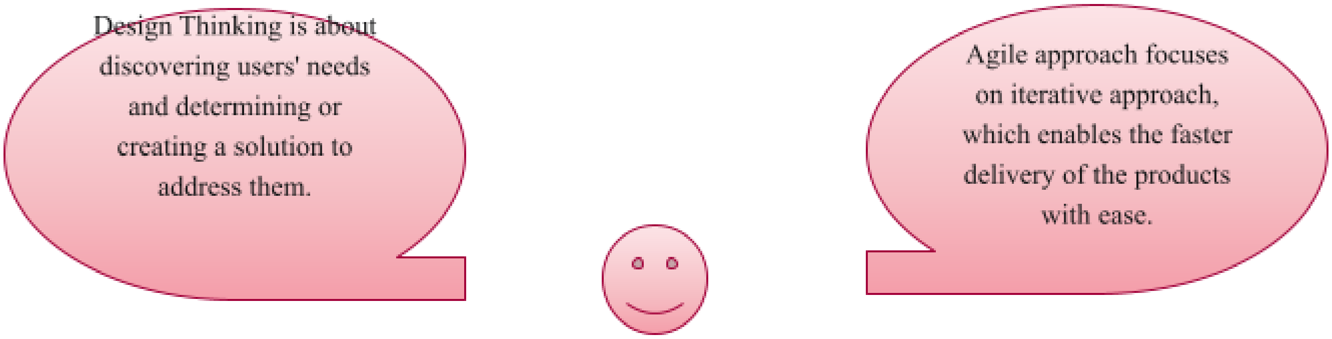
The purpose of this paper is to provide practical solutions for development teams to incorporate Design Thinking principles into different phases of the Agile Software Development Lifecycle (SDLC).

Introduction

Design Thinking has been around in corporate world for a long time, but many companies are not aware that **Design Doing** is essentially equally important as **Design thinking** for producing best solutions. In many companies embracing Design Thinking, the emphasis is placed on thinking part, which is tied to observation, analysis, and research, and very little on the doing part, which is about applying the insights gained to produce actual creative solutions. Design Thinking is most associated to strategy, where we use designing methods to find the right question and begin to answer it. Agile on the other hand is more associated with engineering where once a question is asked, teams iterate toward a solution. *For teams to produce cutting-edge high-quality products or services, they need contributions and control over both strategy and engineering.*

Why Together?

Design Thinking is about discovering users' needs and determining or creating a solution to address them. It's different from just "Design" because it's focused on exploration, creating concepts and ideas for both new and existing products and services. Design Thinking helps teams to better understand their end-users by focusing on why they do what they do and how software can help them do it. Agile is about delivering design solutions by focusing on small incremental design decisions that can deliver the most value possible to the market quickly. It is also about taking information from market delivery and adapting the entire implementation process. Agile allows teams to make truly empowered design decisions based on market feedback. Together, design thinking and Agile create a user-centric environment focused on rapid, frequent iterations as a means of reaching optimal outcomes. Use design thinking to identify the right problems to solve, and then use Agile to iteratively build solutions to solve those problems.



Design Thinking is about discovering users' needs and determining or creating a solution to address them.

Agile approach focuses on iterative approach, which enables the faster delivery of the products with ease.

Synergy between Agile & Design Thinking

So, is there anything that is common between both?

Yes, let's look at the below table to understand:

	Emphasis is on people and collaboration instead of process.	**Iterative or Ongoing Refinement via feedback loops and channels.**	**Empathy is the key that puts "individuals and interactions over processes and tools".**
Agile	Revolves around different ceremonies such as retrospectives, stand-ups, and planning.	Focuses on delivering the MVP (Minimum Viable Product) via development iterations.	Plays a pivotal role to ensure not just a good environment but in return making sure team is building the product the end user will want.
Design	Collaboration is achieved though bringing different groups of people having varied specializations and backgrounds together to resolve a problem.	Works alongside feedback loops throughout the discovery or design SDLC phases.	To ensure human-centric products are designed, follows processes such as user journeys, empathy mapping, and rapid prototyping.

Hurdles faced while using Agile & Design Thinking

Aren't they two separate things:

Yes, let's look at the below table to understand:

	Agility Level	**Process Management**	**Problem**
Agile	The feedbacks or learning received in previous iterations are often put into backlogs and don't get implemented until the next iteration.	The software can be deployed at any point and is often released in alpha, beta, or other stages, in return offering more opportunities to obtain and implement feedback.	Works best when problems are defined and known for building solutions.
Design	Design Thinking has a fail-fast approach as its often executed in earlier phases of SDLC processes, its often easier to adapt or shift gears because of feedbacks for teams.	Design works as a distinct project process with its own beginning, middle, and end, separate to the Agile software development process.	Works best when problems are not known.

Combine Agile and Design Thinking

It’s suggested that to earn better solutions using Design Thinking and Agile together, one should focus on low-risk, high-value opportunities. And with better results, move on to take more challenging initiatives.

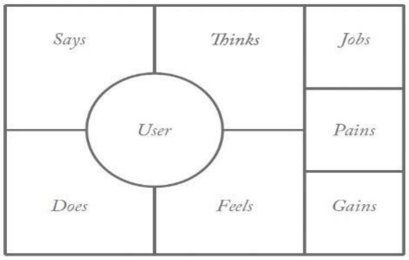
How, lets provide you an answer as follows:

Pick small problems!

- In case of any existing data, starting by testing and working around some small problem.
- Moving on to the design thinking part by building a map of the user's journey.
- Which in turn encourages team members to focus on empathy phase and discovering new solutions.
- Ensure the entire design team understands the end-users.

Understanding User: Power of Empathy Maps!

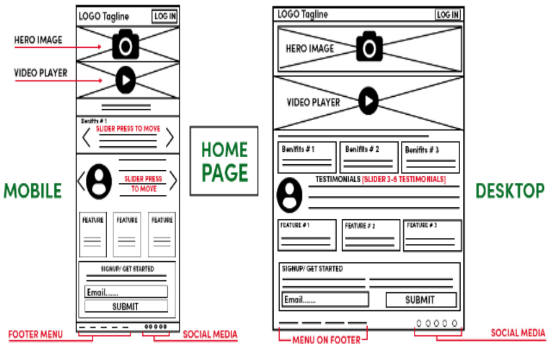
By definition “Empathy mapping is a way to characterize your target users to make effective design decisions. It is just a tool used to articulate what you know about a particular type of user - their needs, goals, expectations, behaviors, pain points, etc.” It is typically achieved as part of the Discovery or Inception Phase.



- Have an Empathy Map Template
- Development Team, Product Owner, and other key stakeholders who understand the end-user’s needs.

Work Early on Rapid Prototyping!

Prototyping is nothing new but has recently gained traction due to its increased use in Agile design methodologies and its best practices that focuses on human-centric designs and principles. It is an exploratory method which allows the team to visualize, explore, design and test ideas in minimum time and cost. Just like Empathy Mapping, Prototyping is generally completed in the Inception or Discovery phase but can also be performed during the Development to design, visualization, and testing technical designs phase. There are multiple approaches and tools to facilitate the prototyping session and highlights as to how to use low fidelity sketches. Low fidelity sketches are rough drawings that are usually completed on paper and are great for an internal team discussion and self-brainstorming sessions.



Maintaining Good User Experience!

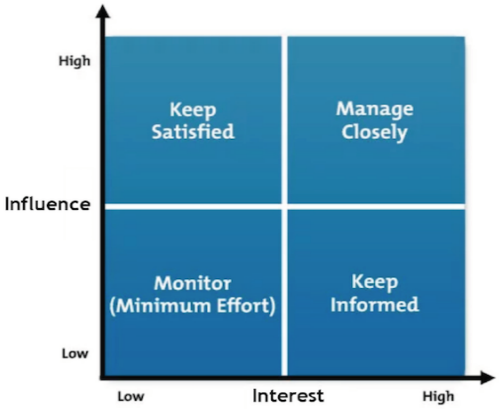
This stage helps to minimize design and development time. The Design patterns are the building blocks that play a pivotal role in encouraging team members to eliminate lower-level design decisions. However, the created patterns should be easily implemented and accepted by the entire team.

Get the right people on team!

Even if everything is well planned and designed, nothing will work if you don't have the right team. Form a core design and development team. The team should consist of decision-makers, architects, visual designer, UX researcher, scrum masters, developers, and QAs. Do not exceed the team to more than 10 members and ensure that every professional has equal involvement and say in discussions. Create a healthy environment that supports collaboration across departments, such as a successful design solution.

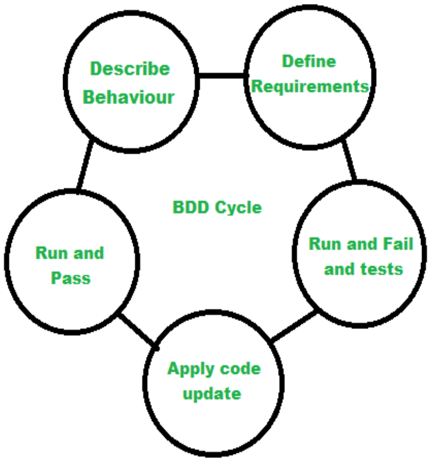
Identify, classify and Work strategically with all stakeholders!

Stakeholder Mapping offers a way to teams to "identify all of the players involved in complex interactions and to understand the means for which [the players] engage and communicate." It's also used to determine when input or participation is required from a set of stakeholders and to what extent. To conduct your own Stakeholder Mapping session, generate the Stakeholder Mapping template, create the development team, and appoint the product owner.



Once, all stakeholders are categorized, lay down a communication plan for each stakeholder class. The communication plan can be as minimal as the Working Agreement that comprises the following:

- Type(s) of Communication Channels that will be used (e.g., Slack, emails, JIRA tickets) to interact with each group.
- The information format and frequency of communication (e.g., weekly/monthly status reports, JIRA dashboard, sprint showcase)
- How/when/what type of feedback is collected, and by whom.



Create Behavior Driven Development (BDD) User Stories!

The Behavior Driven Development (BDD) originates from Test Driven Development (TDD) and adds additional value to TDD tactics. "BDD focuses on the behaviors of your system exhibits, rather than the implementation details of it." Once the team is formed and design has started to take form, start creating the user stories around it.

While creating the BDD user stories follow the format below:

Title: The title should define an activity and act as a headline to define what action needs to be taken and by whom, and for what.

As a <*end user*> I need or want <*feature or ability*> so that <*benefit*>.

Background: The background should be a story style segment that explains the reasons (business case) the feature is being requested at the high level. A good background will narrate:

- Where the request comes from and what is driving the request (e.g., when did a specific event take place?),
- What will be the effect of implementing the request (e.g., referring the example above, users will save time when searching), and
- What will be the consequences of not employing the request?

- **Definition of Done (DOD):** The DoD section should define the high-level acceptance criteria that needs to be fulfilled for the product owner to sign off on a feature or ability.
- **Scenario(s):** Each story should include at least 1 scenario in the Given-When-Then format. These scenarios must serve as testing acceptance criteria. An ideal scenario format is shown below:
  - Given <*some context or a brief description*> when <*I do something*> then <*this happens*> when <*I do another thing*> then <*this new thing happens*>.

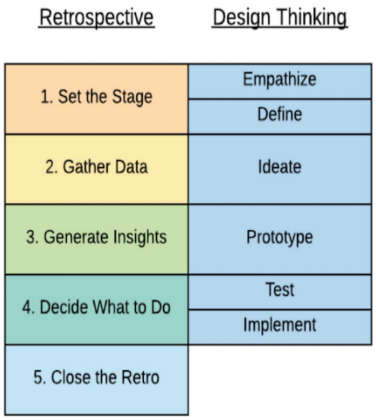
Conduct Periodic Testing!

Set up a testing schedule, based on characteristics of the project. The time can vary and can be either scheduled once in few days or once during the sprint. Understand the viability of ideas during the early stages and test simple prototypes to eliminate errors. Test the software and evaluate the result for the best output.

Conduct Design Thinking Centric Retrospectives!

Design Thinking can also be easily incorporated into a team's regular retrospective ceremonies. By navigating a team through a Retrospective that focusses on defining and empathizing critical problems, a team can be better equipped to identify actions for testing and implementing for resolving problems. You can use the below steps to incorporate design thinking into your next retrospective.

1. **Empathize:** As a team, focus must be on brainstorming the list of challenges or problems the team experienced while finishing the sprint or release.
2. **Define:** Collaboratively select the most significant point for the retrospective.
3. **Ideate:** All ideas are good ideas. Work around as many ideas as possible to solve the problem or challenge sans judgment.
4. **Prototype:** Choose a solution and spend time discussing it and come up with a solution blueprint.
5. **Test & Implement:** Create a test plan to test out the solution to avoid breakdowns. It helps in determining what essential inputs are needed to make the solution happen. The test plan should also include how and when to perform the test or implement blueprint.

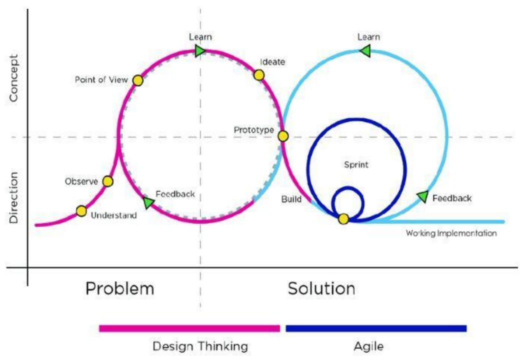


Retrospective	Design Thinking
1. Set the Stage	Empathize Define
2. Gather Data	Ideate
3. Generate Insights	Prototype
4. Decide What to Do	Test Implement
5. Close the Retro	

Our Success Story…

Our team was given a task to improve the user experience of customers using a tool which was recently acquired by our client's company, it was of high value due to its rich client base. We used used design thinking with agile to deliver the tool with high value addition!

- *We started with multiple knowledge transfer sessions where our team engaged in understanding the existing system, the problems our client were facing and what were the expectations for modernization of tool. We used Product Demo and User interviews with Questionnaires. We used MIRO board for collecting all pain points and expectations.*
- *We used empathy maps to understand our users' mindsets. We added putting additional sub-sections "Pains" and "Gains" to the Think & Feel area on the Empathy Map template.*
- *We identified and classified our stakeholders and came up with different personas. We used these personas throughout to ensure everyone participates freely.*
- *We started thrashing high level product ideas on MIRO for brainstorming sessions with our users. We started collecting answers together for our existing pain points. For example, the team explored persona's actions towards changed UI design for a specific feature, how the different actions will be performed in new design and how data flows will be handled in future.*



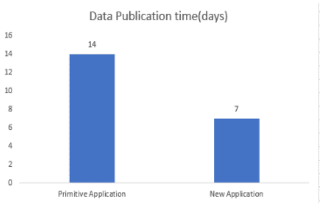
- *We produced low fidelity prototype designs on MIRO board with our designers. We performed periodic testing of our understanding and idea feasibility with our users and team members frequently to adopt and incorporate early feedbacks.*
- *Once the designs were approved, we build the product backlog and defined the MVP scope of our product and started putting priorities to all epics for building timelines and product roadmap.*
- *We built the right team with right capabilities. Product and Testing team started early on BDD framework. The user stories and acceptance criteria were produced in a way that automation could be done by tester on the new tool from first release. All setup was done on JIRA.*
- *We build and deploy the product in increments with most value and less defect there by helping the users to gain maximum benefit from our releases.*

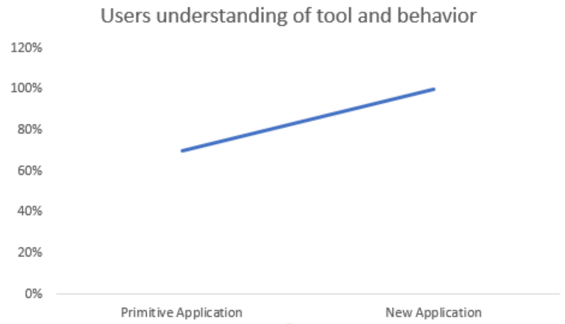
The key to our success was due to following reasons:

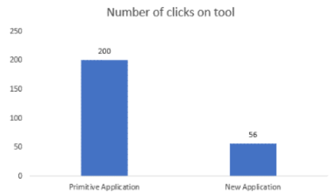
- We had the courage to fail early and learn from our mistakes during workshops!
- We tried to make it new for our users and did not emphasize on making only better!
- We let the ideas flow freely in the team and set up a good culture which helped ideas turn to actions and measurable results!
- We balanced the ideas and actions during implementation to ensure progress!
- We controlled and planned our work in sprints to reduce spillovers!
- We performed retrospection of our actions at every stage!

- We solved problems together and creatively!

Success was measured:







Conclusion

In today’s fast-paced and highly competitive world, the pressure to deliver products that users will love has dramatically increased. But this crazy speed has taught companies how to use design thinking and be Agile. Thanks to these two methods, we can now discover the problem and solve it as quickly as possible.

- Design Thinking is a creative and practical way to resolve the problem quickly. It does that by channeling teams through a human-centric design process which allows for cooperative and iterative approach. Having the ability to truly understand the user’s needs does have a deep impact on a team’s ability to produce high-quality products and services that people would like to use.
- Most problems and challenges that a team faces during the development process be quickly recognized and solved during Design Thinking process. Design Thinking permits the teams to fail quickly in a safe environment and making sure the team evolves further sooner with minimal risks.
- Many Design Thinking practices and principles can easily be applied and/or adapted throughout the entire Agile SDLC lifecycle as there are multiple ways to integrate Design Thinking into the Agile development process.

If you want to create an innovative solution and deliver it fast to the market, design thinking and Agile development are meant to be together for the sake of your product. The tactical activities mentioned in this white paper are only the tip of the iceberg.