How Agile Influences Business

by Glorium Technologies

What Is Agile

Agile development is an umbrella concept for software development methods that are designated to promote tightly knit teams, interactive development, early delivery, continuous improvement, and instant response to the changing requirements.
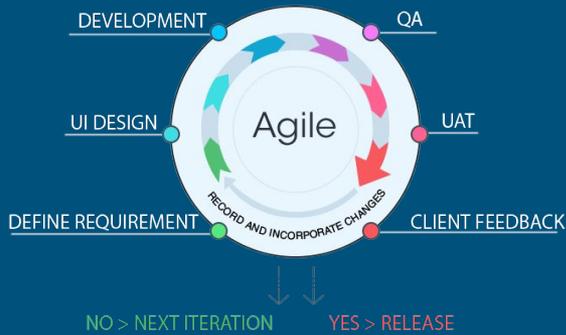
12 Principles of Agile

1. Satisfy customers with the early and continuous delivery of valuable software is the highest priority.
2. Changing requirements should be welcomed, even if they are late. Changes are good if they provide customer's competitive advantage.
3. Frequent working software delivery, max a couple of months; shorter timescales are preferable.
4. Business and IT units must work together in scope of the whole project.
5. Engage motivated people to the project. Provide them with trust and support.
6. Consider face-to-face conversation as the most effective way of information sharing.
7. Working software as the main progress measure.
8. Constantly maintain sustainable development with all the participants of the development process.
9. Technical excellence and good design enhance agility.
10. Take it simple and minimize the amount of work done.
11. Self-organizing teams raise the best specialists.
12. Team regularly assesses the progress and decides how to make it more effective with further iterations.

One of the agile approach fundamentals is that business does not know about all its needs and requirements up front. Often, there is no comprehensive understanding of what the system should do at the very beginning of the software process. Agile accelerates the integration of newcomers into the projects.

Planning/Feedback Loops

Release Plan
Months
Iteration Plan
Weeks
Acceptance Test
Days
Stand Up Meeting
One day
Pair Negotiation
Hours
Unit Test
Minutes
Pair Programming
Seconds
Code

Agile Approaches

There are four main agile approaches: extreme programming (XP), scrum, lean software, and crystal.

XP is an Agile methodology with a high level of involvement between the customer and the development team.

The customer drives development output by prioritizing the most important functions as user stories. The team delivers the user stories iteratively through continuous programming, testing, planning and close collaboration. Working software is delivered very frequently, typically every 1 to 3 weeks. Agile adopters outline high collaboration, minimal documentation, and a fast and aggressive delivery model. However, it requires dedicated people outside the IT department and everyone involved to know XP. There is a shortage of documentation for handover.

Best suited for smaller development teams consisting of senior developers with excellent communication skills who are able to work with and manage non-IT people.

DEVELOPMENT QA

UI DESIGN Agile UAT

DEFINE REQUIREMENT RECORD AND INCORPORATE CHANGES CLIENT FEEDBACK

NO > NEXT ITERATION   YES > RELEASE

CRYSTAL methods focus on people, interaction, community, skills, talents, and communications. While important, process itself goes after.Crystal methods dictate frequent delivery, reflective improvement, close communication, personal safety, focus, easy access to expert users, technical environment with automated tests, configuration management, and frequent integration.

Crystal is very tolerant to the fact that teams can achieve success in different ways, so it is easy to implement. It promotes free speaking of team, so that speaker won't be ridiculed for any thoughts.

LEAN SOFTWARE applies lean manufacturing method to the software development process. Lean manufacturing aims at minimizing waste without sacrifice of productivity. The main focus is to emphasize what gives value and reduce everything else, defined in the following principles: eliminate waste (muda: extra features, task switching, partially done work, ineffective communication), amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity in, see the whole.

To the benefits of lean software development belong overall efficiency of the development process and its speed-up, early delivery of the project. On the other hand, the team working in compliance with lean software should be extremely cohesive; success relies on the team's discipline; software specifications can evolve (not stable).

Scrum

takes a more broad-brush approach to building software than XP.

It is a framework for managing and controlling iterative work at the project level. In SCRUM, a "product owner" works with both business and IT teams to identify and prioritize system-wide functionality in the form of a "product backlog" (requirements list of the software developed).

Various team members sign up to deliver shippable increments of working software in what is known as a sprint, which generally lasts up to 30 days.

Scrum provides high level of project control, active teamwork and transparency. Business gets direct access to the development and its progress and can be adaptive. Can sometimes lead to the "fast and dirty programming".

Best suited for product-focused IT departments that are to deliver a broad scope of products in tight collaboration with project owners.

Agile Statistics

67%

is constituted by companies that either have implemented the pure Agile or are leaning toward it.

More than a half of IT companies are now using agile. On the other hand, not all of these companies fully implement the agile methodologies. One of the reasons for this might be the lack of universal definition of agile.

For instance, scrum and XP sometimes comprise agile methodology. Large enterprises are the most active adopters of agile approaches. As a result, the belief that agile is not for huge and critical projects has been dispelled. Different agile approaches can manage teams working in different time zones.

60 percent of companies surveyed state that less than half of their teams implement agile. 80 percent are in the middle of maturing the level of agile. Despite all this, 98 percent succeeded with agile.

Agile has turned out to be the most popular in such industries as **High Tech, discrete manufacturing, real estate, banking and finance**.

The majority of companies reported to use agile methodologies. 67 percent constituted companies that either have implemented the pure Agile or are leaning toward it.

pure waterfall
2%

leaning torward waterfall
7%

pure agile
16%

leaning toward agile
51%

hybrid
46%
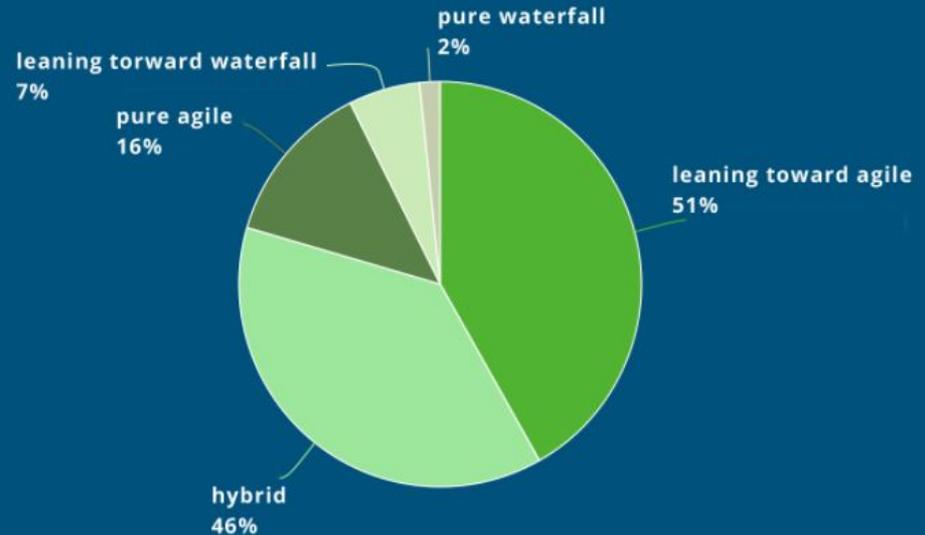
Key motivation for the companies to switch to agile were:

Enahnance collaboration between teams that don't usually work together	54%
Increase the level of software quality	52%
Increase customer satisfaction	49%
Shorten time to market	43%
Reduce cost of development	42%

The majority of respondents mentions three main benefits of the agile:

- effective management of changing requirements and priorities,
- increased team productivity,
- improved project transparency.

Half of the respondents use Agile when outsourcing their projects.

Measuring is an important part of any process. Companies prefer to measure agility with:

- the on-time delivery,
- business value,
- customer and user satisfaction,
- product quality,
- product scope.

The most popular ways to measure projects led with Agile are:

- velocity,
- iteration burndown,
- release burndown,
- planned vs actual stories per iteration,
- burn-up chart,
- Work-in-Process.

The top means to scale agile are mentioned as follows:

- executive sponsorship,
- consistent process and practices,
- common tools across the whole team,
- agile consultants and trainers.

Challenges and obstacles when implementing Agile:

- company culture and philosophy,
- lack of experience with agile,
- lack of management support,
- organization resistance,
- lack of business customer product owner.

Among methods and practices used in the scope of Agile, Scrum is an unconditional leader, followed by Scrum/XP Hybrid and Custom Hybrid approaches.

The most popular Agile tools currently in use are taskboard, bug tracker, Kanban board, spreadsheet, Agile project management software, Wiki, Unit test tool, continuous integration tool (frequent automated code comprising to quickly define troubles with integration).

Business can benefit from Agile providing the following statements are true:

- Customer requirements change fast.
- Customer delivers feedback quickly and is ready to collaborate tightly.
- Team faces complex problems without clearly defined scopes and is ready to produce creative breakthroughs with a tight cross-functional collaboration.
- Team can break work into parts with iterative cycles.

Examples from the real life

How agile has influenced your business

Raphaela, Marketing team lead, has noticed many benefits after Agile integration:

- It is easier for me as a team lead to keep everybody's focus aligned with our overall strategic goals and to ensure that our resources are used for those tasks that have the highest priority.
- Our communication both within the team, cross-departmentally and with management has improved and become more structured.
- Team members seem more motivated and ultimately more productive now, which I attribute to the fact that we have started thinking in two-week sprints. This provides us with a tangible goal that we can work towards and look forward to, rather than just swim in a sea of never-ending tasks and projects. Agile coach, Tushar, mentioned that business became flexible, receptive to feedback, and nimble in the overall structure.

According to **Steve**, a software developer, it is about increasing value generated by delivering the right stuff and addressing issues sooner because of the additional feedback, visibility and steer-ability and because the people doing the work are empowered to make real-time, common sense, "tactical" decisions instead of waiting for information to go up and down the management chain.

Maik, an author of The Agile Manifesto Unfolds, said that Agile improved his business with better flexibility (less work in progress with more feedback cycles) and better quality (less bugs that you build new features on).

What goals did you expect to achieve with Agile?

With Agile, **Chuck**, Agile Project Management Author and Instructor, was trying to achieve an adaptive approach, well-designed for dealing with projects that have a high level of uncertainty.

Andre , Agile coach and trainer, wanted to provide consistent delivery of value to the customers with Agile. His main objective was happy customers and happy teams.

What kind of team structure and hierarchy do you have?

Fernando with 11 years of experience in development and lead roles, have seen many ways of a team organization. Usually, there are project managers to care of the more bureaucratic, business and financial aspects, like contract negotiation, personal development and major conflict resolution, for example.

There is not much hierarchy, except for occasional team leaders, who perform most of a scrum master's role, keeping up with the team's activities and sending out status reports and so on. These may or may not share the technical leader role. They may also step in to address specific, timely demands from clients, like prioritizing a proof of concept or extracting an one-time report from the database.

The team members themselves are self-managed when it comes to actual project development: they show up for scheduled meetings, they estimate using the agreed methods, prioritize the sprint work items, assign themselves tasks to do, submit and review each other's changes to source code, trigger and handle quality activities, suggests improvements on the process, etc. Those teams consist mainly of developers, testers and designers.

Release plans are presented by team leaders and approved by project managers and product owners, but only after having their feasibility attested by the team members. Hierarchy is not really as important as commitment is, while respect and open communication are keys to making functioning teams.

What kind of team structure and hierarchy do you have?

Leon, an Agile Delivery Lead, uses several approaches to his team structure, namely:

- cross functional teams, i.e. a small group of people who, collectively, possess all the skills and knowledge to produce valuable, tested software;
- as little hierarchy as possible, within reasonable constraints of basic corporate governance, operational risk, etc.;
- empowered people, i.e. each team is largely left alone to decide how and what to build. Decades of research tells us that people are the most motivated and perform the best when they are given "Mastery, Autonomy and Purpose" (from Dan Pink's book "Drive", read it NOW if you haven't yet)".

For **Ravneet**, a founder of Agile school, flat hierarchy works the best. As more levels of hierarchy lead to more delays and more bureaucracy. However, flat hierarchy works best when talking about startups or small organizations. As the organization grows, it becomes difficult for flat hierarchies to sustain.

The structures that he has seen to be working best in big organizations are to have a flat hierarchy within one product. This keeps the delays and bureaucracy to minimum and innovation to maximum.

What Agile methodology is the best to choose?

Chuck has chosen an adaptive approach that fits the methodology to the nature of the problem rather than force-fitting all problems to some predefined methodology.
He believes that within Agile, there really isn't a lot of choice:

- Scrum is by far the most widely-used methodology for projects
- Kanban is by far the most widely-used for non-project work

An important thing to understand is that Agile methodologies such as Scrum and Kanban are intended to be very adaptive. For that reason, many people consider Scrum to be more of a framework than a methodology.

Another important consideration in selecting a methodology is the team and the organization's readiness to implement it. It would be very difficult, if not impossible to implement any Agile methodology if the team is not well-trained and the organization is not well-aligned with implementing it.

The most important thing to save from here is that you should apply and test as many methodologies as you can. Then see what suits better in your environment and team. And if possible make an all-in-one cocktail with your favorite pieces from multiple methodologies.

Does your team like to work with Agile?

Adam, Customer Success Manager, says that his team love to show an increment (progress) on each review meeting, gathering feedback from their stakeholders, and improving the teamwork on retro. Working in sprints is a very good idea. You can divide a project into small, manageable parts. But the best part is transparency and visibility of their daily work.

Tiago, Head of Product at OLX, also answers positively: "They are not the only ones liking, our stakeholders like as well, since the team are delivering (great) value with a good rhythm".

Peter Kristensen, IT teamlead, sees agile as a great way to challenge bad decisions and get the whole team involved in making better decisions. His main obstacle was the very beginning: "After a while the team figured out that they were no longer evaluated individually by the number of tasks each one handled (and therefore mostly took easy tasks), but by value of the tasks that the team produced.
After a while he asked how they felt about it:

- "To begin with it was frightening. Suddenly it was up to us how to solve a problem. But after a while we found that it was much more exciting to work this way"
- "More fun ... and I think we have made better solutions"
- "Confusing when no one told us exactly what to do, but I like that now"
- "We are working together now instead of competing to look good individually"

Does your team like to work with Agile?

Tushar's team likes Agile. He came to the conclusion through such metrics:

- Attrition rate has decreased
- Employee satisfaction has upward swing
- Most of our new hires are from referral
- Glassdoor rating has upward swing
- Overall effectiveness has increased many folds. Consequently, productivity and efficiency are increasing.
- Employees spend more time on value creation instead of repetitive tasks (they automate repetitive tasks)

What biggest challenges have you faced while integrating Agile?

As a challenge, **Tiago**, outlines the team maturity level, environment, and industry. However, the most usual challenges that he has faced were the following:

- Team rhythm adaptation to the new methodology while it still needs to keep the product development rolling out.
- Lack of procedures.
- Lack of structured planning and organized meetings.
- Difficulties on timing/effort estimation when doing the spring planning (low level of experience/maturity inside the team).

The solution or answer for most of all these issues is Time and Routines.

Andre, Agile coach and trainer, defines culture as the main challenge. There are companies where hierarchy has been established for a long time. As a result, people are used to commanding and getting commands.

Abram, healthcare software development manager, encountered such challenges as culture and specific industry. The former came from insufficient support from upper management and buy-in from the key stakeholders. The later is derived from the regulations and cost present in the specific industry. A short budget cannot afford "fail early, fail often". It's "measure twice, cut once". There are downturns & margin level with very small budget variance allowed, and where everything has to have corresponding processes and documented for compliance.

What major benefits and disadvantages have you found in Agile?

Implementing proper procedures (e.g. daily stand up meetings every day at the same time, no excuses) makes the team getting into routines and healthy habits. This increases the team rhythm. And consequently, motivation, because the results will be visible. Daily stand up meetings help everyone have a big picture of what is being developed and what the blockers are.

By defining a sprint time box duration (only to help you know when it starts and finishes), at the end of the sprint you can put your team inside a room and discuss the feelings of the past sprint in order for all of you to have a sense of accomplishment and discuss what should you continue doing (the positive things), what to stop (the wrong things/mistakes) and what to start (to be improved/wish list).

Jaimi McEntire, a software development manager in Jack Henry & Associates, defines such benefits as predictable development, constant discussions, less misunderstanding, and constant review if the development effort matches the business needs.

The major disadvantages are uneven sprint, not-balanced load. At the beginning of a sprint, the QA team does not have enough work, at the end of the sprint, the development team is winding down, and the QA team has too much work. There are of course ways to address this - the development crew tests at the end, and the (canonical) suggestion is for QA to work on "test scripts" at the beginning. But these have their own issues.

Because a team has to bid to complete items, and the sprint is seen as unsuccessful if they do not complete the items, there is a tendency to underbid and sandbag (letting the work fill the time), leading to a decrease in productivity. The constant review and discussion give some developers the feeling that they are being micro-managed.

What major benefits and disadvantages have you found in Agile?

Lisa, a founder, sees Agile's benefit in its bias toward action. Work tends to expand to fill the time allotted so with shorter cycles there's more motivation to continuously get something out the door. The built-in cycles of planning, doing, and reflecting aid the creation of shared habits, so more time can be spent on continuous improvement and less time on meeting logistics.

Cross-functional teams are also helpful, and most teams I've seen also included a diverse range of ages and backgrounds. A wider range of perspectives can help spot issues and opportunities to better serve customers and the enterprise.

A disadvantage is when agile teams over emphasize the short-term, so teams can end up on a treadmill of sorts, always looking only at the next few weeks. From her experience, that tends to come from leaders seeing agile teams as code manufacturing units and not design teams. By focusing on throughput, they're incentivized to minimize team involvement in anything that won't go out the door right away. That means complex and creative projects get pushed down the backlog.

There can also be an aversion to any type of planning or architecture. It's true that the future is uncertain and plans will likely change, but in my experience, the lack of a draft vision, roadmap, transition states, and integration plan ends up costing more money due to delays, confusion, defects, business process issues, and customer complaints than the time it would take ahead of time to just jot down a shared plan. Especially when you're looking to use agile to transform a legacy environment and you need to coordinate with multiple other teams to make it happen.

What major benefits and disadvantages have you found in Agile?

Kirk Bryde, Agile coach, Telstar Agility, believes that the main Agile's benefits are:

- Giving customers what they really need (not just what they say they want).
- Faster time-to-market (by incremental and iterative releases - each one getting the customers a little closer to what they really need).
- Higher s/w quality (through automation, and by avoiding bloated releases).
- More predictable release schedules (by fixing the schedule and varying scope).
- Happier customers (by delivering usable s/w early and often, and by keeping customers constantly aware of the status of their projects and products via frequent demos of new features, and by letting the customers drive the product roadmap).

According to Tiago, to the other benefits belong:

- increased odds of producing software of good maintainability;
- focus on addressing business goals and actual user needs;
- proper systems engineering discipline
- promoting open culture.

What metrics do you use to track Agile efficiency?

Allen, Agile trainer, believes that the business has to change to accommodate the needs of Agile processes and thinking. At the business level, what matters is agility. The business has to be able to deliver a change at the same pace that they discover the needs for them. To do that, they need to eliminate everything that slows things down. That can involve everything from introducing Agile processes to complete reorganization. The only metric needed is how fast you can get a change into your user's hands (and whether the users are delighted with the change).

The metrics should be in tracking how well the company is doing at achieving its goals instead of intermediate measures that assume a particular way of achieving those goals. Let the people do the work and be responsible for how they choose to work. Agile is just a means for achieving company goals by facilitating the people doing the work taking full responsibility for how they work.

Alan, Senior Agile Coach and Certified ScrumMaster, sees Agile as about seeking value, about effectiveness. The real value in Agile ways of working is found when we focus on effectiveness.

Kirk Bryde, Agile coach, Telstar Agility, says that the best metrics are from direct customer feedback. There's also some leading indicators that can show the likelihood of any trailing indicators (like customer feedback).

Paul Oldfield, a team member in Youmanage, believes that rapid implementation of a new strategic direction should be used as a metric: how fast a company produces values or product and how fast it can respond to a change.

What metrics do you use to track Agile efficiency?

David, Agile coach, advises to measure the current lead time from when the business discovers it needs to change until the change has been made and is being consumed by customers. Anything which reduces that lead time increases agility. You need to continually ask if you reduce the time to implement a change from the time the business realizes the need to change? To look at the cycle time from discovering the need to when the change is made and users are benefiting from the change can be also used as a metric.

Allen, principal in Holub Associates, thinks that speed is a factor. Business agility should be seen as the ability to implement change at the rate that the need for the change discovered. That's the ideal, at least. Most businesses can't do that, so there will be some time slippage. They'll discover a need, and the change will take longer to do than it might in an ideal world. As one approaches the ideal, that slippage reduces to 0.

What metrics do you use to track ǝ Agile efficiency?

Steve Gordon, a software developer and Agile coach believes that Success criteria and progress should not be on KPIs or Agility, but only on the delivery of quality and value to customers.

Where quality or value delivery is problematic, then root causing the problem would get into KPIs and Agility to determine how to address the problem and see if remedies are working.

Constantly measuring anything besides progress on the end objectives will inhibit the self-organization, authority and responsibility of the teams, which will in turn inhibit Agility.

Rob Walker, a software development director, mentioned that Agile allows business to be lean, and deliver valuable products with little or no waste. The best metric is validated learning. This is obtained by making small testable assertions, testing / validating them quickly, and pivoting with some new, small, testable assertions if necessary.

Conclusion

Agile's goal is to make business processes efficient, result-oriented, and tolerant to changes.

1. People actively adopting Agile have outlined the following benefits of Agile: better flexibility, less work in progress, more feedback cycles and constant review, better quality, healthy habits, increased team rhythm, visible results, increased motivation, predictable development, constant discussion, less misunderstanding, continuous improvement, real value for customers, faster time-to-market, focus on business goals, open culture promotion, and project transparency.

2. As the main goals they defined achievement of adaptive approach, improvement of collaboration, upgrade of quality of soft delivered, customer satisfaction, shorter time-to-market, decrease of cost, and consistent delivery of value.

Conclusion

The most useful, Agile will be for the companies where requirements change fast, customer delivery is quick, problems are complex, scope is not defined, there is a need of creative breakthrough, collaboration is tight and cross-functional.

3. The most popular structure and hierarchy inside Agile team turned out to be cross-functional teams, little/flat hierarchy, empowered people, and respectful and open communication.

4. The most popular methodologies are Scrum, Kanban, and all-in-one cocktail of methodologies.

5. The most challenging were the rhythm of adaptation, lack of procedures, lack of structured planning, time planning in estimation, company culture, absence of product owner, and lack of management support.

Conclusion

6. Agile adopters see the main disadvantages in uneven sprints, not-balanced load, and underbidding.

7. The main metrics to measure Agile success are speed of change, goals achieved, customers feedback and satisfaction, rapid implementation of new strategic directions, value generated before and after Agile, on-time delivery, product quality, efficiency and Work in Process.