



Agile Capability Inventory

Based on the 12 principles of the agile manifesto

For instructions to perform this assessment see:

<https://sprintagile.com.au/maturity-assessment/>

Agile Principle 1. Why does your team exist? Why do you come to work every morning? Is it to deliver features for the product? Is it to generate profit for the company? Your highest priority must be delivering value to your customer and delighting them.

8. The existential purpose of our team is to deliver value to our customers and delight them.
5. We have more conversations about delighting our customers and value, than delivery timelines, scope, and features.
3. We have more conversations about delivery timelines, scope, and features, than delighting our customers and value.
2. Customer delight and value is of low priority for us. Timelines, scope, and delivery are more important things.
1. We don't even register on the radar.

Agile principle 2. Do you welcome changing requirements?

8. We thrive on change. When our customer changes their mind our eyes spark and we get a dopamine boost.
5. We understand that changing requirements is natural, but it still causes a minor inconvenience for us.
3. Our technology stack and delivery commitments make it difficult for us to welcome changing requirements once we have started working on something.
2. Only if our customer could make up their mind and tell us exactly what they want early on, we could deliver much faster and life would be much easier.
1. We don't even register on the radar.

Agile Principle 3. How often do you deliver working software to production (could be feature toggled off)?

8. Usually on daily basis.
5. At least once every 2 weeks.
3. At least once every month.
2. Less than once a month.
1. We don't even register on the radar.

Agile Principle 4. How closely do business people (product owner, business or customer representatives) and developers (development team members including programmers, BAs, testers, UX, and so on) work together in your team?

8. What are you talking about? There are no distinctions between business people and developer for us. Our business people are developers, and our developers are business people. When Elon Musk was coding PayPal, was he a developer or a business person?

5. Business/customer representatives work with developers on a daily basis and have casual conversations on a daily basis.

3. Business/customer representatives attend the daily stand-up meetings every day. But apart from the official stand-up meeting, they only have casual conversations with developers every few days.

2. Business/customer representatives only attend the important meetings such as sprint planning and sprint reviews (showcases).

1. We don't even register on the radar.

Agile Principle 5. Which one describes your team?

8. We are a 100% autonomous team; we just do things and then tell our leadership we've done it. Normally the leadership is stunned how creative we are and how we came up with a solution better than they would have even imagined. Sometimes we make mistakes, but our leadership is supportive and helps us learn from our mistakes. No idea is too crazy or too silly.

5. We have a great degree of autonomy. When we come up with creative ideas and decide to implement them, we run those ideas past our leadership and tell them we intend to do a certain thing. Our leader helps us see the flaws in our ideas and reduce the probability of mistakes. When mistakes happen, our leader is supportive and helps us learn from our mistakes. Some ideas are considered too crazy or too silly.

3. Creativity is encouraged but when we make mistakes, we get frowns. Typically, people think twice before proposing a creative idea as they don't want to be seen as a stupid person. We need to take creative ideas to our leadership and get approval from them before starting to work on them. Our leadership scrutinises the idea before making the go/no go decision.

2. Even though the words "creativity" and "innovation" is used across the company, we are not comfortable coming up with creative ideas, let alone implement them autonomously. Our environment is risk averse and we have to always be super careful not to make any mistakes. There are consequences for mistakes and this fear of making mistakes means we can't experiment with creative, new, and unproven ideas.

1. We don't even register on the radar.

Agile Principle 6. When you have a question from someone, or you want to inform someone of something, what is your preferred approach in most cases?

8. I walk to their desk and ask them. Or I call them with a video conferencing tool (with video on). I often use whiteboards, virtual whiteboards, or screen sharing, to facilitate the conversation. If needed, after the conversation I send an email with the summary of what was discussed.
5. I call them or have a teleconference call with them with my video off. If needed, after the conversation I send an email with the summary of what was discussed.
3. Most often I find it enough to chat with them using a messaging service, and I find no need to call them. (This option excludes things that are super quick and simple.)
2. I email them, write a page on the wiki, or comment on the Jira ticket.
1. We don't even register on the radar.

Agile Principle 7. How do you measure our progress?

8. We don't have an explicitly defined end state, nor a detailed scope. Instead, we have clearly defined business outcomes we want to achieve. We measure and track improvements to business metrics that we hypothesised will improve as a result of our work on an ongoing basis as the work is released. Of course, we also demo working product every iteration and seek feedback.
5. We demo our work every iteration and seek feedback from business stakeholders how they feel our work contributes towards the business objectives. Business people know how much more work is left to achieve their desired outcomes. But it is not clear to the team how business people measure progress against the business objectives or metrics.
3. We have a list of features or a defined scope we are working towards. We tick off features from this list when we deliver or demo those features to our business stakeholders.
2. We track progress based on milestones and sign offs of various phases of the project/program.
1. We don't even register on the radar.

Agile Principle 8. How do you keep sustainable constant development speed indefinitely?

8. We don't have a clearly defined end state for our product. We don't have requirements or specifications. Instead, we have a product vision, and our product is going to exist forever. So, we improve our product by moving it towards the vision. The vision also adapts over time slowly as the markets change.

5. The business needs certain things certain times. We negotiate scope with our business people to reduce our timelines while helping achieve their business goals. But occasionally we agree that scope cannot be reduced, and we have to work harder and longer hours for short periods of time to deliver what needs to be delivered.

3. The business gives us the scope and asks us to estimate. When we provide an estimate, and then they influence us to reduce the estimate. Then those estimates are treated as commitments and the business holds us to our estimate. And when inevitably things take longer, we have to work harder for longer hours under stress to be able to deliver the agreed upon scope in the agreed upon dates.

2. The projects are handed over to the delivery team with defined scope and defined time. Estimation and delivery are done by two different groups of people. We have to delivery what is in the specification by a specified date. This means sometimes work is easy and often work is stressful.

1. We don't even register on the radar

Agile Principle 9. How much do you value technical excellence?

8. All our developers have read "Clean code" by Uncle Bob (or similar books) and consider themselves "Software Craftsman". Our developers are highly active in the community and attend dev conferences on a regular basis. There is no compromise on code quality and we always avoid incurring the type of technical debt that makes it harder to code in future.

5. Some of our developers have read "Clean code" by Uncle Bob (or similar books) and are familiar with the ideas behind Software Craftsmanship. We sometimes attend conferences and meetups. We try to write high quality code. Sometimes we have to incur tech debt to release value early but then later we fix that tech debt.

3. What is software craftsmanship? Who is Uncle Bob? We just care about writing high quality code and try to do a good job. Training is mostly organised for us by our bosses. Typically fixing tech debt is de-prioritised but every once in a while, we get to work on fixing the code smells in hackathon like events.

2. We often do a tactical solution instead of a strategic solution. We usually intend to do the right thing and fix the technical debt later but the "tyranny of urgent" gets us often and we have to move to the next piece of work before we can actually fix the code. The code base is becoming like a spaghetti. We have a few "heroes" who have a deep understanding of the code base, but most other developers have to spend a lot of time trying to unweave the complexities of our systems to be able to program new features.

1. We don't even register on the radar.

Agile Principle 10. How do you increase simplicity by maximising the work not done?

8. We actively attempt to minimise the number of features we build. Business people (business/customer representatives) come to us with business problems and we come up with solutions with minimum changes to the product. Our business people believe less features are better and strive for making our products, services, and enterprise architecture, simpler.
5. We like simplicity, but our business people don't believe simplicity can be achieved by reducing the number of features in our product. Our business people believe to remain competitive in current markets we need more features and bigger changes to our product. We typically push back on features and offer simpler solutions to solve the business problems, but sometimes it is hard to convince our business people less features are better.
3. Our executives talk about simplicity, but they try to make things simple mostly by focussing on enterprise architecture. We don't believe less features are better. Practically speaking, no one is pushing back on feature requests. We try to deliver to the requirements that the business provides to us.
2. We actively try to maximise the number of features to our product. There are so many bells and whistles in our product, and we keep adding more bells and whistles to our products and services. More complex products mean happier customers and more revenue for our company.
1. We don't even register on the radar.

Agile Principle 11. How do you come up with architecture, requirements, and design?

8. Architecture, requirements and design are not prepared upfront by specialists. They emerged as the team develops solutions for business problems. Architecture, requirements, and designs, are evolved and concretised by the team as solutions are developed. There are no specialised roles for architect, UX, or analyst in the team but the team has the capability to develop high quality architecture, designs and requirements as per industry standards.
5. The team has specialised skills for architecture, UX, and requirements. Architects, designers, and analysts are embedded in the team and work closely within the delivery team to adapt the architecture, requirements, and designs as the product is developed.
3. Architecture, designs, and requirements are developed outside the delivery teams. But the delivery teams collaborate closely with architects, UX, and product/requirements teams to initially define the artefacts and subsequently evolve those artefacts as the solution is being developed.
2. The Architects come up with architecture, the UX team comes up with designs, the product/requirements team come up with the requirements. And then they are handed over to the delivery team and the delivery teams deliver to those architecture, requirements, and designs. When the delivery team has questions, the external teams may not be available immediately to answer those questions. Sometimes it could take a few days to be able to get the answers we are looking for.



Agile Capability Inventory

Based on the 12 principles of the agile manifesto

For instructions to perform this assessment see:

<https://sprintagile.com.au/maturity-assessment/>

1. We don't even register on the radar

Agile Principle 12. How do you reflect on becoming more effective and adjust your behaviour?

8. Every retrospective we come up with one to three measurable experiments for improving our effectiveness. Then in the next retrospective we re-evaluate the experiments and decide to perish, pivot, or persevere with the experiments.

5. We come up with improvement actions in retrospectives and implement them before the next retro. These are actions, not measurable experiments. Also, we are not sure what is the difference of between efficiency and effectiveness.

3. In our retros we talk about what went well, what was bad, and what to improve. We typically don't get actionable outcomes from the retrospectives.

2. What is a retrospective? We don't run retros, or skip them when we are too busy.

1. We don't even register on the radar.