



Values



Company over team over individual

We value decisions that have a positive impact for the entire tech organization even if they have a negative impact on our own team. We rationally assess the impact and acceptance of both new and current technologies on other teams. We look for opportunities to help other teams benefit from what we are building.



Ease of people movement

The movement of people into teams with complex, high priority work is an important part of how Scout24 will continue to grow as a business. To reduce onboarding barriers, we value standardization of technologies and processes across Scout24. We value engineers who can work across multiple stacks.



Security and reliability

The security of our products, tools, applications and users' data takes our highest priority, followed by its reliability. We consider the criticality and maturity of a product when deciding the level of effort required for resilience. We react with urgency if problems are detected and we apply our security best practices to all our applications.



Full ownership and clear responsibility

We believe teams are responsible for supporting, monitoring, alerting, operating and retiring everything they build and maintain. Responsibilities and ownership should always be clear and published. Everything must be owned.



Moving fast

We optimize for speed of execution and learning over cost efficiency. We believe it is important that teams contain the necessary people and power to build features and products without a dependence on other teams. We use common platforms and proven technologies to free product engineering teams from building infrastructure as a prerequisite to creating value without compromising quality.



Outcome over effort

We celebrate the impact of our work over the effort we've been putting in or the amount of code we produced. We understand the potential impact of our work to our business and search for an easy, fast, secure way to validate our ideas before writing any code. We accept when these goals cannot be achieved, are comfortable moving on and learn from this experience. We reuse existing solutions where possible in order to execute and scale the team quickly.

Principles

Agile software development

We listen to user needs and release value often so we can learn from successes and failures and adjust our approach quickly.

Product first

We focus on the value we are delivering to our users, both internal and external, and avoid expending time and money on technology for its own sake.

Embrace platform model

We make full use of our Scout24 platforms, and view platforms as a collaborative effort between product and platform teams.

Autonomous cross-functional teams

We work in small teams completely responsible for taking product concepts to visible customer value.

You build it, you run it

We, as team members, ensure that the services our team create continue to run reliably, which motivates us to create services that are intrinsically reliable.

Cleaning is part of work well done

We keep our products and their code clean to ensure a productive work environment.

Data is a Shared Asset

We treat data as a shared asset to ensure all permitted staff have a complete view of the company performance.

Continuous integration and delivery

We, as team members, continuously integrate our software changes with those of our teammates and push these changes to production automatically.

Minimize technology variation

We deliberately minimize the variety of technologies in our system to drive efficiency, and, at the same time, have a managed program of innovation.

Automate

We automate repetitive activities to achieve better predictability, efficiency, reproducibility, resiliency, and security.

Eliminate accidental complexity

We are diligently keeping complexity at a minimum.

Apply principle of least privileges

Security is important to us. We segregate permissions and responsibilities in order to reduce the blast radius of security incidents and to demonstrate clear ownership and accountability.

Loosely coupled and highly cohesive services

We strive to minimize the coupling of services and maximize their cohesion in order to, among other things, limit failure propagation, reduce response times and limit the scope of software changes.

