

Our values

Agile Teams

We are solving problems together in self-organizing teams following Agile values and principles. After several years of practicing Scrum and Kanban we are able to make the best of both for particular projects and teams.

Quality as a first class citizen

We are proud of the quality of our products. In 9LivesData Quality Assurance starts from the very beginning of each project. Practices enhancing quality are present in the design, coding and maintaining of our systems. We do not sacrifice quality to meet deadlines.

Proper Process

Different projects require different software development processes. Based on our experience from startups and enterprise products we are able to choose the best approach.

Kaizen

Continuous improvement is our way of working. Each failure is a chance for learning and teams have time for regular inspection and adaptation.



The 9LivesData Way

Our Principles

Self-organization

Teams in 9LivesData are self-organized. They decide how to do the work, what tasks are needed to meet the goal and who should work on which task. Self-organization is complemented by guidelines, best-practices and rules created in 9LivesData during more than 10 years of experience in software development. As in the Scrum framework the team as a whole takes responsibility for the project.

Daily Standup

9LivesData adopted the idea of Daily Scrum and it is used in vast majority of teams. Managers can be invited by the team but It is not for the managers and not for status reporting. It is for the team to synchronize and stay on track.

Small Teams

We prefer small teams working smart on well defined goals. Team of 4 is perfect, team of 6 is acceptable, team of 7 is too large.

Supportive Managers

Managers in 9LivesData are here to help the teams work in the best possible way and let people become better software engineers. They support self-organizing teams by coaching, mentoring and ensuring that 9LivesData way of software development is well understood by the team members. Managers create environment in which Agile values and principles are alive

Whole Team working directly with the client

It should be always considered to engage the whole team in direct work with client representatives or a client team. Detailed discussions - especially technical ones - should be done directly between team members and client representatives.



Our Principles

Product Owner Proxy

When the client is not available daily for the team Proxy Product Owner (PPO) can work for a client and help the team to do the right things. Often she leads planning and demo sessions, accepts the work done and provides sufficient reporting for the client. Usually this role is played by one of the managers.

Transparency

All important aspects of the project are transparent within 9LivesData and for our clients. In particular it includes policies, practices, definition of done, scope, schedule and current progress of the project. E.g. client should be informed about our process (unit testing, code reviews, good design etc.) so that she knows the amount and the quality of finished work.

You never work alone

Working in a team enhances good design and code quality, promotes learning and collective code ownership. We promote teamwork for all types of tasks, including software development and project management. Each person should have a peer for the task who will discuss design decisions, testing approach and perform the code review. Collaborative task decomposition improves completeness and correctness of the design.

Shepherd for new teammates

New team members have one person helping them in the first weeks of work. This person is usually a peer in coding and code review, first source of knowledge regarding team habits and standards.

Openness

In 9LivesData every team member feels comfortable to share his ideas and opinions. New team members are encouraged to propose changes as well as get and give feedback.



Our Principles

Collective Intelligence over Heroes

Best designs and project decisions emerge from collaboration, collective efforts and competition of many individuals. Decisions are made by consensus within the team of professionals.

Pair Programming

We promote pair programming for non-trivial or learning tasks such as difficult bug analysis or bugfixing, coding new modules etc. Pair programming also proved to be a good approach for introducing new team members and spread knowledge within the team.

Technical Excellence

In 9LivesData we all focus on technical excellence and software craftsmanship. We know it is a never ending story of continuous learning and mastering our skills. The best way of learning is by practice and collaboration and that's the way we do it.

Regular Retrospectives

Regular team retrospectives are the key element to enable continuous improvements. Teams meet regularly to discuss and improve their way of working. Retrospective commitments should be explicit to make changes happen. It is ok to conduct retrospective each 2-4 weeks - it's schedule can be bound to iterations schedule, but it is not mandatory. In addition new product version or problem oriented retrospectives are welcome as a good inspection point.

Feature Leader

In large and complex software projects such as as HYDRAsTOR multiple teams are involved and multiple features are implemented concurrently. Large features (weeks of work of the team) are led by the Feature Leaders. They lead functional and technical designs, monitor progress of development and coordinate work if multiple teams are involved, but the design and development remain a collaborative effort. Feature Leader should ensure that the guidelines and rules are followed by the team in the implementation of the feature.



Our Principles

Feature QA Leader

Feature QA Leader supports Feature Leader in terms of ensuring the quality of the feature. She is responsible for the Test Plan and for making sure that all quality activities are finished on time. She should pay special attention for the comprehensiveness and completeness of testing approach and the quality of tests.

Limiting Work in Progress

In 9LivesData we have adopted limiting work in progress to improve the flow of tasks and decrease lead time for the features.

Continuous Progress Monitoring

Projects or features which are expected to take more than 4 weeks are divided into a sequence of Milestones. Each milestone has valuable, achievable and demonstrable goal e.g. "product is used internally by the team" or "it is ready for testing by external performance team". It improves traceability and helps to attack complex problems step by step.

Milestones

Projects or features which are expected to take more than 4 weeks are divided into a sequence of Milestones. Each milestone has valuable, achievable and demonstrable goal e.g. "product is used internally by the team" or "it is ready for testing by external performance team". It improves traceability and helps to attack complex problems step by step.

Short iterations

In 9LivesData iterations are used by the teams for better work organization but are not mandatory as some teams follow continuous flow or Kanban approaches. Iterations are short (1-2 weeks) and always have a clear goal. We prefer clear iteration goal over detailed scope definition.



The 9LivesData Way

Our Principles

Team Estimations

Estimations in 9LivesData are used for progress monitoring and for making conscious decisions about implementation of features or functionalities. We do not use estimations nor velocity tracking for judging team performance. We are aware that even best estimations are in most cases wrong and should be revised when the work progresses. Estimation is made by the developers in a collaborative effort using techniques such as affinity estimation or estimation poker with the preference for simple ones.

Automate (almost) everything

Every work that is repeated should be automated to minimize the effort needed in a long term. This includes build procedures, test running and scheduling, visualization and metrics.

Coding Standards

Coding standards based on the industry standards and team experience are defined and documented for all the programming languages used in 9LivesData. Coding standard rules are enforced by automated checkers and continuous integration routines.

Peer Code Review

Peer code review is a part of the 9LivesData workflow. It is done shortly after coding and supported by automated code review task generation. To make the review efficient code to be reviewed should be committed in small, focused batches. Both production code and test code is reviewed. Code review guidelines help to remember about most important verification points.

Well-fitting Quality

Each project has its appropriate quality level. It may be different for distributed products installed in hundreds of customers all around the world and different in a fast-changing tech startup. In 9LivesData we agree with the client on the proper quality level making sure she is aware of the cost and benefits for proposed quality assurance approach.



Our Principles

Proud of the software quality

We want to be proud of the quality of the software we create. We will not lower the quality even if required by the client. And for any other reason.

Collaborative Designs

Quality starts at the design phase. We promote collaborative, incremental and iterative design done in series of meetings with relevant people. For difficult or cross-component features design is reviewed by external parties like other teams or senior developers.

Comprehensive Unit Testing

Unit testing is an essential part of implementation - our code has comprehensive and sound unit tests created as a part of coding activities. Unit tests quality is as important as the code quality.

Code Coverage by Unit Tests

Code coverage by unit tests is periodically measured and analyzed. The goal of this activity is to detect modules or files in the code which need discussion and possibly improvement in unit testing. Certain products can have clear coverage requirements in their definition of done for releasing the code to the production (e.g. 85% line coverage for each file larger than 20 lines of code).

Continuous Integration

In 9LivesData code is usually co-created by multiple teams and developers. In most cases they integrate the code continuously using automated build system. The production code is kept in good condition. Each time someone pulls out the code from the repository it should compile, pass unit tests and basic functional tests. If those conditions are not met it is quickly detected and authors are informed. Continuous Integration (CI) in 9LivesData is achieved by many means i.e. automated builds, unit testing, automated functional testing and use of CI Servers.



Our Principles

Story Branches

Some teams follow pull-request model for short-term development. It helps to isolate changes and improves code review process, but requires discipline and short features.

Avoid Long-Lived Feature Branches

Long-lived feature branches and late merging are discouraged. It can be considered in special cases like large and disruptive features but it always has to be discussed before such decision is made by the team. In such case feature branch should be regularly re-based to the master (which is assumed to be in good condition - see Continuous Integration).

Pre-release Tests

To keep the main codeline in good condition each push to the repository is tested before it's accepted. Testing includes unit testing and selected set of functional tests (so called pre-release tests) which ensure that basic product functions are not damaged. Pre-release testing should be reasonably fast to make development comfortable.

Test Plan as part of the Design

Test plan for a feature complements the design and enhances team knowledge about what should be done and how should the system behave after the feature is ready. Valid test plan should include all relevant aspects of testing of a feature, including common and corner cases, performance, load tests etc. It should be prepared in parallel or just after the design is finished to enable Acceptance Test Driven Development. Best test plans are usually created under a supervision of the QA Leader.

Legend

Rule Should be followed by the team. In special cases can be omitted for good reasons.

Practice The team can decide to apply this practice, but it is not mandatory.

Culture Something you should feel.