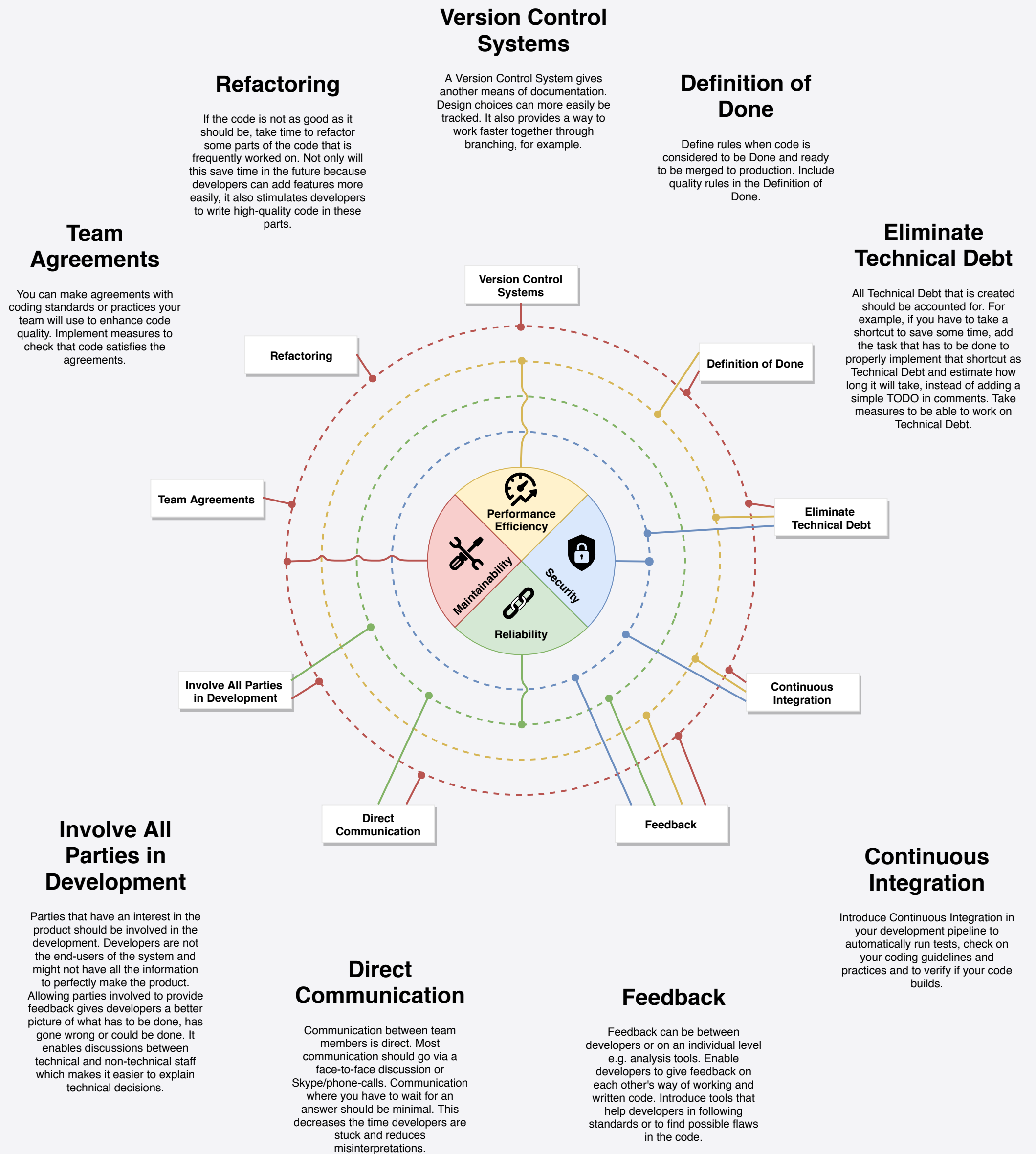# Improving Code Quality in Agile Software Development

Insights into how you can improve your team's code quality metrics through enhancing your Agile processes

Version Control Systems

A Version Control System gives another means of documentation. Design choices can more easily be tracked. It also provides a way to work faster together through branching, for example.

Refactoring

If the code is not as good as it should be, take time to refactor some parts of the code that is frequently worked on. Not only will this save time in the future because developers can add features more easily, it also stimulates developers to write high-quality code in these parts.

Definition of Done

Define rules when code is considered to be Done and ready to be merged to production. Include quality rules in the Definition of Done.

Team Agreements

You can make agreements with coding standards or practices your team will use to enhance code quality. Implement measures to check that code satisfies the agreements.

Eliminate Technical Debt

All Technical Debt that is created should be accounted for. For example, if you have to take a shortcut to save some time, add the task that has to be done to properly implement that shortcut as Technical Debt and estimate how long it will take, instead of adding a simple TODO in comments. Take measures to be able to work on Technical Debt.

Central diagram

- Performance Efficiency
- Security
- Reliability
- Maintainability

Outer labels: Version Control Systems, Refactoring, Team Agreements, Involve All Parties in Development, Direct Communication, Feedback, Continuous Integration, Eliminate Technical Debt, Definition of Done

Involve All Parties in Development

Parties that have an interest in the product should be involved in the development. Developers are not the end-users of the system and might not have all the information to perfectly make the product. Allowing parties involved to provide feedback gives developers a better picture of what has to be done, has gone wrong or could be done. It enables discussions between technical and non-technical staff which makes it easier to explain technical decisions.

Direct Communication

Communication between team members is direct. Most communication should go via a face-to-face discussion or Skype/phone-calls. Communication where you have to wait for an answer should be minimal. This decreases the time developers are stuck and reduces misinterpretations.

Feedback

Feedback can be between developers or on an individual level e.g. analysis tools. Enable developers to give feedback on each other's way of working and written code. Introduce tools that help developers in following standards or to find possible flaws in the code.

Continuous Integration

Introduce Continuous Integration in your development pipeline to automatically run tests, check on your coding guidelines and practices and to verify if your code builds.