

Crystal Methods

From Wikiversity

Contents

- 1 Introduction
- 2 The Crystal Family
- 3 Commonality
 - 3.1 Frequent delivery
 - 3.2 Reflective improvement
 - 3.3 Close or osmotic communication
 - 3.4 Personal safety
 - 3.5 Focus
 - 3.6 Easy access to expert users
 - 3.7 Technical environment with automated tests, configuration management, and frequent integration

Introduction

Crystal methods are a family of methodologies (the Crystal family) that were developed by Alistair Cockburn in the mid-1990s. The methods come from years of study and interviews of teams by Cockburn. Cockburn's research showed that the teams he interviewed did not follow the formal methodologies yet they still delivered successful projects. The Crystal family is Cockburn's way of cataloguing what they did that made the projects successful.

Crystal methods are considered and described as "lightweight methodologies". The use of the word Crystal comes from the gemstone where, in software terms, the faces are a different view on the "underlying core" of principles and values. The faces are a representation of techniques, tools, standards and roles.

Methodology, techniques, and policies are differentiated between by Cockburn:

- Methodology - set of elements (e.g. practices, tools)
- Techniques - skill areas (e.g. developing use cases)
- Policies - dictate organizational musts

Crystal methods are focused on:

1. People
2. Interaction
3. Community
4. Skills
5. Talents
6. Communications

Cockburn says that Process, while important, should be considered after the above as a secondary focus. The idea behind the Crystal Methods is that the teams involved in developing software would typically have varied skill and talent sets and so the Process element isn't a major factor.

Since teams can go about similar tasks in different ways, the Crystal family of methodologies are very tolerant to this which makes the Crystal family one of the easiest agile methodologies to apply.

In his research, Cockburn [1999], he defines behaviour of people in teams:

- “People are communicating beings, doing best face-to-face, in person, with real-time question and answer.”
- “People have trouble acting consistently over time.”
- “People are highly variable, varying from day to day and place to place.”
- “People generally want to be good citizens, are good at looking around, taking initiative, and doing ‘whatever is needed’ to get the project to work.”

The points above are why Crystal methods are so flexible and why they avoid strict and rigid processes typically found in older methodologies.

The Crystal Family

Cockburn developed the different methods in the family of methodologies to suit teams of different sizes which need different strategies to solve diverse problems.

The Crystal family of methodologies use different colours to denote the “weight” of which methodology to use. If a project were a small one a methodology such as Crystal Clear, Crystal Orange or Crystal Yellow may be used or if the project was a mission-critical one where human life could be endangered then the methods Crystal Diamond or Crystal Sapphire would be used.

The family is divided into different colours. Some examples:

1. Crystal Clear
2. Crystal Yellow
3. Crystal Orange
4. Crystal Orange Web
5. Crystal Red
6. Crystal Maroon
7. Crystal Diamond
8. Crystal Sapphire

	Clear	Yellow	Orange	Red	Maroon
Life (L)	L6	L20	L40	L80	L200
Essential Money (E)	E6	E20	E40	E80	E200
Discretionary Money (D)	D6	D20	D40	D80	D200
Comfort (C)	C6	C20	C40	C80	C200
	1-6	7-20	21-40	41-80	81-200

[Source: A Practical Guide to Seven Agile Methodologies Part 2, (<http://www.devx.com/architect/Article/32836/0/page/2>)]

The Crystal Family of Methodologies. One characteristic of Crystal is its intentional scaling to projects based on size and criticality. The larger a project gets (from left to right), the darker the colour.

Commonality

Between all the methods in the Crystal family, there are seven prevailing common properties. Cockburn found that the more of these properties that were in a project, the more likely it was to succeed.

The seven properties are:

1. Frequent delivery
2. Reflective improvement
3. Close or osmotic communication
4. Personal safety
5. Focus
6. Easy access to expert users
7. Technical environment with automated tests, configuration management, and frequent integration

Frequent delivery

Frequent Delivery is the regular releasing of iterations of the software program. This idea comes from agile methodologies. Designers and developers decide what features to include in each release and they design and test for each release. With Crystal methods, iterations are to be released weekly up to quarterly – the release times are dependant on the length of the project.

By releasing iterations, stakeholders will be able to spot problems earlier in the project which will save a lot of hassle later on. Another point on this is that if the end users decide that the project does not do things the way they'd like it to be done, then steps can be taken to resolve this before it is too late.

With Crystal methods, there can be more than one iteration in a release. This is because it may not be feasible to release every iteration, so a collection of iterations are gathered and delivered in a single release.

Reflective improvement

Reflective improvement involves developers taking a break from regular development and trying to find ways to better their processes. Iterations help with this by providing feedback on whether or not the current process is working.

With Crystal methods, the idea of teams holding “reflection workshop” meetings every couple of weeks is encouraged. These workshops help find processes that are and aren't working well and help the team to modify them so that a strategy can be developed that works well for the team.

Close or osmotic communication

In Crystal Clear and the smaller of the Crystal methodologies, Osmotic Communication is used. Osmotic communication involves the team being together in a room and getting information to flow around it. With regards to larger teams (over 8 or so), where distraction can arise, Close Communication is used.

The team must be in the same room for this to work. This is because if the developer has to break concentration to move somewhere else to ask a question then their thought process will probably be lost.

By using this type of communication, information flows quickly throughout the team. Questions can be rapidly answered and all the team members know what is going on as well as having the ability to correct any misconceptions that may arise.

By listening to the others in the team, a developer can pick up on what the others are doing, gain experience and develop new ideas. Developers working near each other can help with problems that the other is encountering.

Communication overhead is greatly reduced by using this type of communication. The need for email updates, extra documentation, etc is lowered. By having the team together, each member knows what the others are doing so they should be able to take over their team-mate's parts of the project if needed.

Personal safety

This has got to do with the issue of free speaking within a group of people. If a person is ridiculed whenever they ask a question, suggest an idea, etc then they will be less likely to speak up the next time. The people in the team must be able to trust each other and feel free to speak up about issues or whatever arises.

Focus

Focus in crystal refers to two things; firstly focusing on an individual task in a project for enough time that progress will be made and secondly, it refers to the direction of which the project is heading.

With the first, the flow of progress is taken into account while thinking of issues that would affect it such as interruptions, meetings, long questions, phone call, etc. It can take a while to get back into the flow again so this delays completion of the current task.

Crystal defines two rules for dealing with issues that may interrupt focus. One is to set a two-hour period where the developer is to have no interruptions. The other one is to assign a developer to a project for at least two days before being switched to another project.

With the second meaning of focus, issues such as definition of goals are discussed. The definitions should be clear and the developers should know exactly what the goals of the project are. The project leader should prioritise the goals which will allow developers to focus on particular areas.

Easy access to expert users

This involves the developers working with a person of expertise in the project area so that the expert can answer any questions, suggest solutions to problems, etc. The expert user should be an actual/real-life user and not just a tester from the development team. The more involved the expert user is (in practical terms), the better since they would have more hands-on experience.

The more time that the expert user can give will greatly help the overall project but this is not always feasible. There should be a minimum of a once a week, two-hour meeting with the expert user, and the ability to make phone calls to the expert user too.

Technical environment with automated tests, configuration management, and frequent integration

The idea behind this is that there should be continuous integration and testing so that if any changes are made, then errors, breakages, etc can be spotted. Since this is done on a regular basis, problems are less likely to grow as they can be resolved earlier in the project.

The process of checking-in code into a repository can help in identifying the problem code and remove it by reverting back or updating with correct code. This is where configuration management and automated testing come in. These allow for quicker unit testing and eventual resolution of problems.

[hide] Software engineering

Computer · Computer science · **Crystal Methods** · EPET · EPETO · Software · Template:Software engineering · Unsolved problems in software engineering

Retrieved from "http://en.wikiversity.org/w/index.php?title=Crystal_Methods&oldid=1021908"



- This page was last modified on 30 March 2013, at 02:10.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy.