

Crystal Methodology

Introducing Crystal

- The crystal metaphor.
 - Crystals have different hardness and color
 - Software projects have size and criticality
- Crystal is a family of methodologies with a copy and alter paradigm versus a kit paradigm.
- Three levels of listening and learning.
 - Create a script
 - Understand changes
 - Zen master
- Apprenticeship is the best way to learn.
- Software is a finite goal seeking cooperative game of invention and communication.

Key Success factors

- Project Manger, Team Lead, Executive (2 of the 3 must be magical)
- Communication and Community
- Good people are everything
- Team lead must know how to tweak the methodology for each specific project
- People trump process, politics trumps people, people working together trumps working alone
- Good people must also communicate
- Citizenship and apprenticeship learning

Benefits

- One methodology does not fit all. There is no one solution
- Avoids the RUP marketing problem, that sells one solution to CEO's
- Crystal is made for high turbulent environments
- Allows people to be people
- Adequate software production is the goal
- Not highly disciplined like XP and PSP. This is realistic.
- Fewest set of rules possible
- Advocates professional social scientists and facilitators to assist
- Address communication in detail. Cold vs. hot for different situations
- Absolute communication is impossible.
- Bottom line: Crystal is realistic

When to use/When to Avoid

- When to Use
 - Anytime you can use Orange, Clear, or Orange Web
 - Up to 60 or so developers
 - Essential, discretionary monies, or loss of comfort
 - Good people
- When to Avoid
 - Life critical systems
 - Developers are not co-located
 - More than 60 developers
 - When you can not use Orange, Clear, or Orange Web
 - If you're a beginner with no real leadership - do XP

Challenges

- Errors on the side of leaving things out that you must add
- Effective communication is very difficult
- In tuning the methodology it is common for project managers to believe they have come up with the one answer
- Tuners tend to embellish methodologies with too many rules, practices, and ideas
- You must be able to spend the time in the planning games and/or interviews to understand
- Tuning a methodology to be intolerant of peoples individual styles
- Not putting the right personalities in the right roles
- Rewarding the wrong things and getting behavior you do not desire as a result

13 Elements of a Methodology

- Roles - take personality into account
- Skills - training, practice, and natural talent
- Teams - organize the team structure
- Techniques - not a focus of Crystal
- Activities - planning games, refactoring
- Process - how the activities fit together
- Work Products - create templates
- Milestones - where are you on the road
- Standards - coding standards
- Quality - how people take pride in work
- Team Values - how the team works together
- Tools - not a focus of Crystal
- Personality - effects everything

Seven Principles

- Face-to-face
- Weight is costly
- Heavier methodologies for larger teams
- More ceremony for criticality
- More feedback and communication
- Fewer intermediate deliverables
- Discipline, skills, documentation