



# Software Development Methodologies

Lecturer: **Raman Ramsin**

## Lecture 12

### Agile Methodologies: Crystal



# Crystal

- Introduced by Cockburn as a family of methodologies in 1998.
- New members of the family were defined in 2001 and 2004.
- Based on the belief that different projects call for different methodologies
- Flexibility addressed through continuous review and revision of the development process



# Crystal Methodologies: Project Categories

- Projects are categorized according to the *criticality* of the system being produced and the *size* of the project.
- Four levels of criticality have been defined, based on what might be lost because of a failure in the produced system:
  - Comfort (C)
  - Discretionary Money (D)
  - Essential Money (E)
  - Life (L)
- The maximum number of people that might have to get involved in a project is regarded as the measure of the project's size.
- A category L40 project is a project involving up to 40 people developing a life-critical system.



# Crystal Methodologies: Complexity

- Crystal methodologies put heavy emphasis on communication among people involved in the project:
  - Projects with a larger size require heavier (more complex) methodologies, since they involve more people, and hence, need better coordination.
  - Projects with higher criticality call for a more rigorous approach, which might be accommodated by tuning a methodology used for a less critical project.
- Crystal methodologies are categorized according to the project size that they address.
- Each member of the Crystal family has been assigned a colour showing its relative complexity: the heavier the methodology, the darker the colour assigned to it.

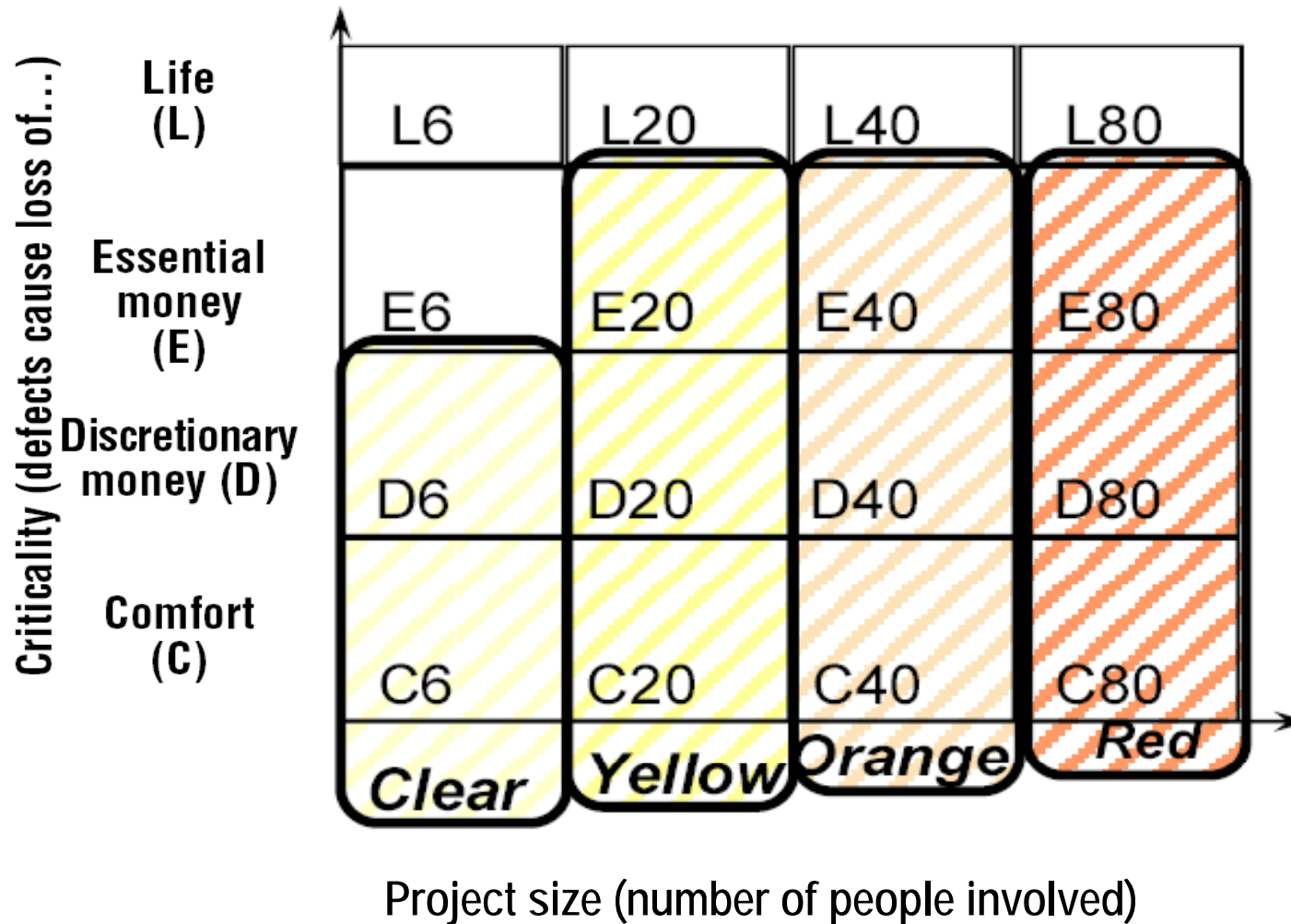


# Crystal Methodologies: Family Members

- Crystal methodologies named in the literature: *Clear*, *Yellow*, *Orange*, *Red*, *Maroon*, *Blue*, and *Violet* (in ascending order of complexity).
- Others can be added if a usage context arises.
- Only those that have been practically used in real projects have been defined:
  - **Crystal Orange** was introduced in 1998, targeting C40, D40 and E40 projects.
  - **Crystal Orange Web** was introduced in 2001, and is a variant of Crystal Orange targeting ongoing web development projects.
  - **Crystal Clear** was introduced in 2004, primarily targeted at C6 and D6 projects.



# The Crystal Family



[Cockburn 2004]



# Crystal Methodologies: Common Traits

- Adherence to the principles of agile development
- Iterative-incremental process, with each increment (delivery cycle) lasting no more than four months
- No support for development of life-critical systems
- No support for distributed teams
- The people involved must be collocated (e.g. in the same building).
- Dependence on effective communication and information flow among team-members for successful enactment



# Crystal Methodologies: Flexibility

- Every Crystal methodology:
  - Enforces a development process framework.
  - Requires that a set of certain general process elements be used.
  - Requires that certain work products be produced.
  
- But a large body of finer-grained detail is left to the development team to decide; developers are even allowed to use techniques borrowed from other methodologies.
  - The development team(s) selects a base methodology at the start of the project (in the form of a minimal set of working conventions).
  - *Reflection Workshops* are frequently held to monitor and tune the process.





# Crystal Clear

1. Primarily targeted at C6 and D6 projects.
2. There is only one development team.
3. Team-members work in close proximity to each other.
4. Usable software is delivered at least once every three months, though delivery is typically expected to be much more frequent.

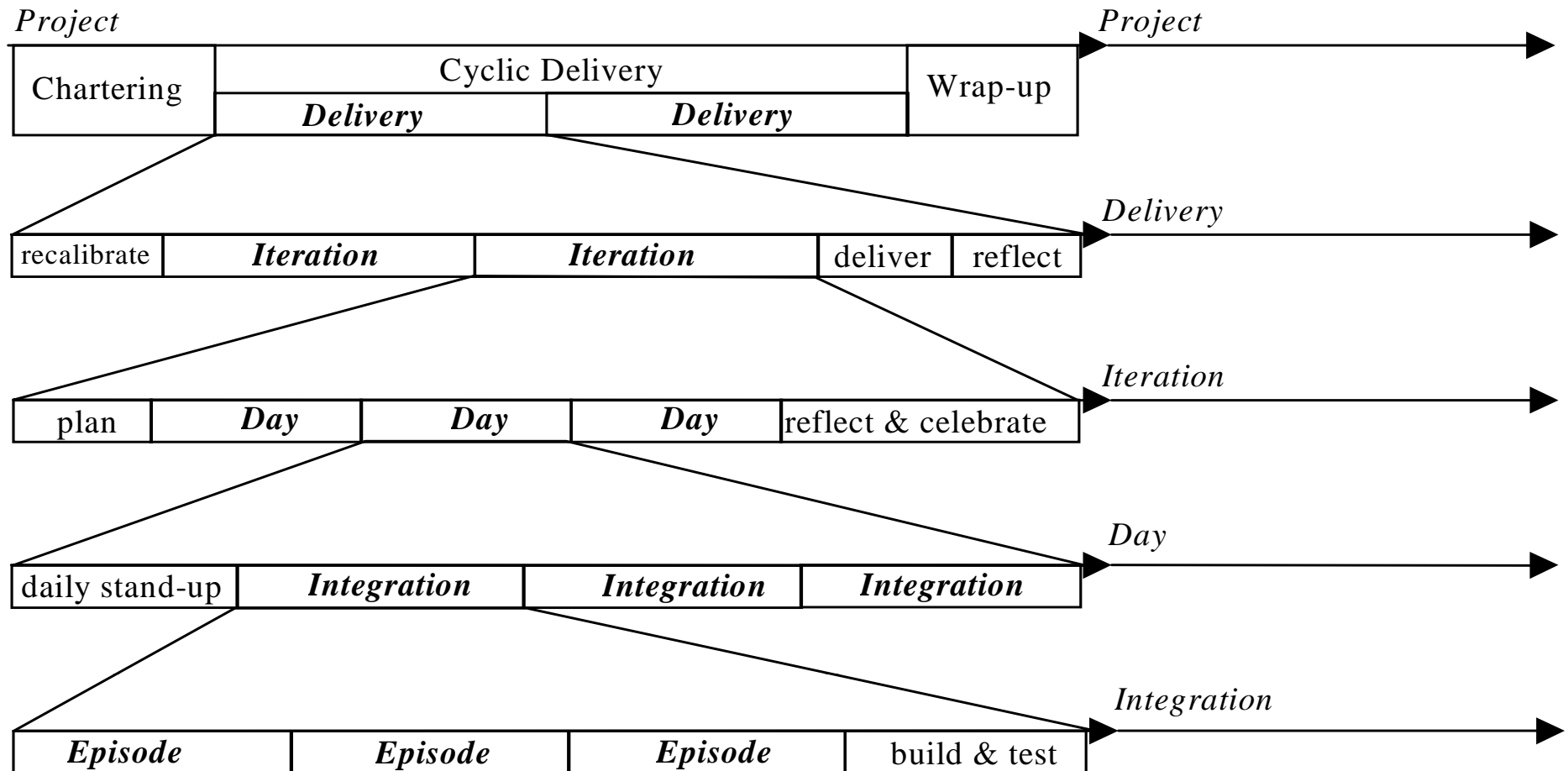


# Crystal Clear: Process

1. *Chartering*: forming the development team, performing a preliminary feasibility analysis, shaping and fine-tuning the development methodology, and developing an initial plan.
2. *Cyclic Delivery*: the main development engine consisting of two or more *Delivery Cycles*. Each delivery cycle takes from one week to three months, during which:
  1. the team updates and refines the release plan.
  2. implements a subset of the requirements through one or more program-test-integrate iterations; the iteration(s) in a delivery cycle are themselves composed of *daily* and *integration* cycles.
  3. delivers the integrated product to real users.
  4. reviews the development methodology adopted and the project plans.
3. *Wrap-up*: deployment into the user environment, and post-deployment reviews and reflections are performed.



# Crystal Clear: Process



[Cockburn 2004]



# Crystal Clear Process: Chartering

- 1. Build the core of the team*
- 2. Perform the **Exploratory 360°***
- 3. Shape and fine-tune the methodology conventions*
- 4. Build the initial project plan*



# Crystal Clear Process: Chartering – Activities 1

## 1. *Build the core of the team:*

1. An *Executive Sponsor*: provides monetary and logistical support and essential direction to the team; may also act as domain expert.
2. A *Lead Designer*: also acts as project manager, coordinator, and technical expert and trainer.
3. An *Ambassador User*, who acts as the expert on system usage.
4. A number of *Systems Analysts, Designer-Programmers, Business Experts, Testers, Text-Writers, Coordinators*, and others, as deemed necessary by the team.

## ~~2. Perform the *Exploratory 360°*~~

- ~~3. *Shape and fine-tune the methodology conventions*~~
- ~~4. *Build the initial project plan*~~



# Crystal Clear Process: Chartering – Activities 2

1. *Build the core of the team*

2. *Perform the **Exploratory 360°***: a preliminary feasibility study providing a high-level project-wide review of the key issues governing the development effort:

1. expected business value of the system
2. high-level requirements (usually captured in use cases)
3. domain models
4. technology alternatives
5. overall project plans and constraints
6. necessary resources
7. development methodology

3. *Shape and fine-tune the methodology conventions*

4. *Build the initial project plan*



# Crystal Clear Process: Chartering – Activities 3

1. *Build the core of the team*

2. *Perform the **Exploratory 360°***

3. *Shape and fine-tune the methodology conventions:*

- A minimal set of rules is agreed upon by the team as the skeleton of the methodology to be used in developing the system.
- The initial set will be iteratively revised and perfected during cyclic delivery, gradually evolving into a methodology tailored to fit the project in hand.

4. *Build the initial project plan*



# Crystal Clear Process: Chartering – Activities 4

1. *Build the core of the team*
2. *Perform the **Exploratory 360°***
3. *Shape and fine-tune the methodology conventions*

## 4. *Build the initial project plan:*

1. typically includes a *Project Map* showing the development tasks and their dependencies, and a *Release Plan* showing the projected completion dates for delivery cycles and iterations.
2. Tasks are identified, prioritized and estimated using a technique called *Blitz Planning*, which is a close variant of XP's card-based planning technique.





# Crystal Clear Process: Cyclic Delivery

1. Each delivery cycle involves the following four activities:
  1. *Recalibrate the release plan*: the requirements and the project plans are reviewed and updated according to the experience gained in the delivery cycles performed so far.
  2. *Develop in iterations*: one or more iterations are performed in every delivery cycle. Each iteration lasts from one week to three months.
  3. *Deliver to real users*: the integrated system is delivered to a small number of users and feedback is used for improving the system and revising the plans and/or the requirements.
  4. *Reflect on the delivery*: through a workshop, the team reflect on the quality of the delivered product, the development methodology and the plans.



# Crystal Clear Process: Cyclic Delivery - Iteration

1. *Iteration planning*: a fine-grained plan is produced involving the tasks that should be performed in the iteration.
2. *Cyclic program-test-integrate*: An iteration consists of cyclic daily activities. A *Daily Cycle* includes:
  1. A stand-up meeting;
  2. Several *Integration Cycles*; during each integration cycle:
    1. Designer-programmers perform design-implementation *Episodes*: they start development tasks, and carry out designing-programming and unit testing.
    2. At the end of an integration cycle, the code produced by designer-programmers during the episodes is integrated into the system, and appropriate integration tests are performed.
3. *Iteration completion ritual*: a Reflection Workshop is held.



# Crystal Clear Process: Wrap-up

1. Perform acceptance testing.
2. Prepare the final product and the user environment for final deployment.
3. Carry out system conversion.
4. Perform final reflection aimed at compiling and recording the lessons learned from the project, in order to use them in future projects.



# Crystal: Strengths and Weaknesses

## ■ *Strengths*

- Iterative-incremental process
- Continuous integration
- Iterative development engine governed by planning and reviewing
- Flexible and configurable process
- Methodologies used for a low-criticality project can typically be tuned to fit a higher-criticality project, provided that the project size is not increased dramatically.
- Active user involvement



# Crystal: Strengths and Weaknesses

## ■ **Strengths** (Contd. 1)

- Early and frequent releases
- Scalability (though limited) through using different methodologies for different project sizes
- Continuous validation
- Specific work-products prescribed, though details and templates are left to the developers to decide
- [Crystal Clear] Traceability to requirements (though limited) through continuous validation and quality reviews
- [Crystal Clear] Requirements are allowed to evolve over time
- [Crystal Clear] Test-based development



# Crystal: Strengths and Weaknesses

## ■ **Strengths** (Contd. 2)

- [Crystal Clear] Preliminary feasibility analysis conducted as a risk mitigation mechanism
- [Crystal Clear] Based on system functionality, typically captured in use cases
- [Crystal Clear] Based on structural modeling of the problem domain
- [Crystal Clear] Based on a system architecture identified and refined during the process
- [Crystal Clear] Design activities encouraged, with results documented as *Design Notes*



# Crystal: Strengths and Weaknesses

## ■ *Weaknesses*

- Only limited scalability
- Lack of an unambiguous common process
- Limited applicability: not suitable for developing highly critical systems
- Over-dependence on inter-human communication
- [Crystal Clear] Seamlessness not addressed
- [Crystal Clear] traceability to requirements suffers because planning and development activities are not necessarily requirements-based (e.g. *Blitz Planning* is task-based rather than requirements-based).



# Crystal: Strengths and Weaknesses

## ■ *Weaknesses* (Contd.)

- [Crystal Clear] Design activities are carried out by individual developers in the manner they choose; design is not performed as a team effort with globally available results based on which implementation can be carried out uniformly.
- [Crystal Clear] Since the detailed nature of many work-products is left to the individual developers to decide, behavioural and functional modeling can be poor throughout the process.
- [Crystal Clear] No formalism





# References

- Cockburn, A., *Surviving Object-Oriented Projects: A Manager's Guide*. Addison-Wesley, 1998.
- Cockburn, A., *Agile Software Development: Software through People*. Addison-Wesley, 2001.
- Cockburn, A., *Crystal Clear: A Human-Powered Methodology for Small Teams*. Addison-Wesley, 2004.