Crystal Methods

A family of Agile Methodologies

ESOF 3MIEIC2 group A:

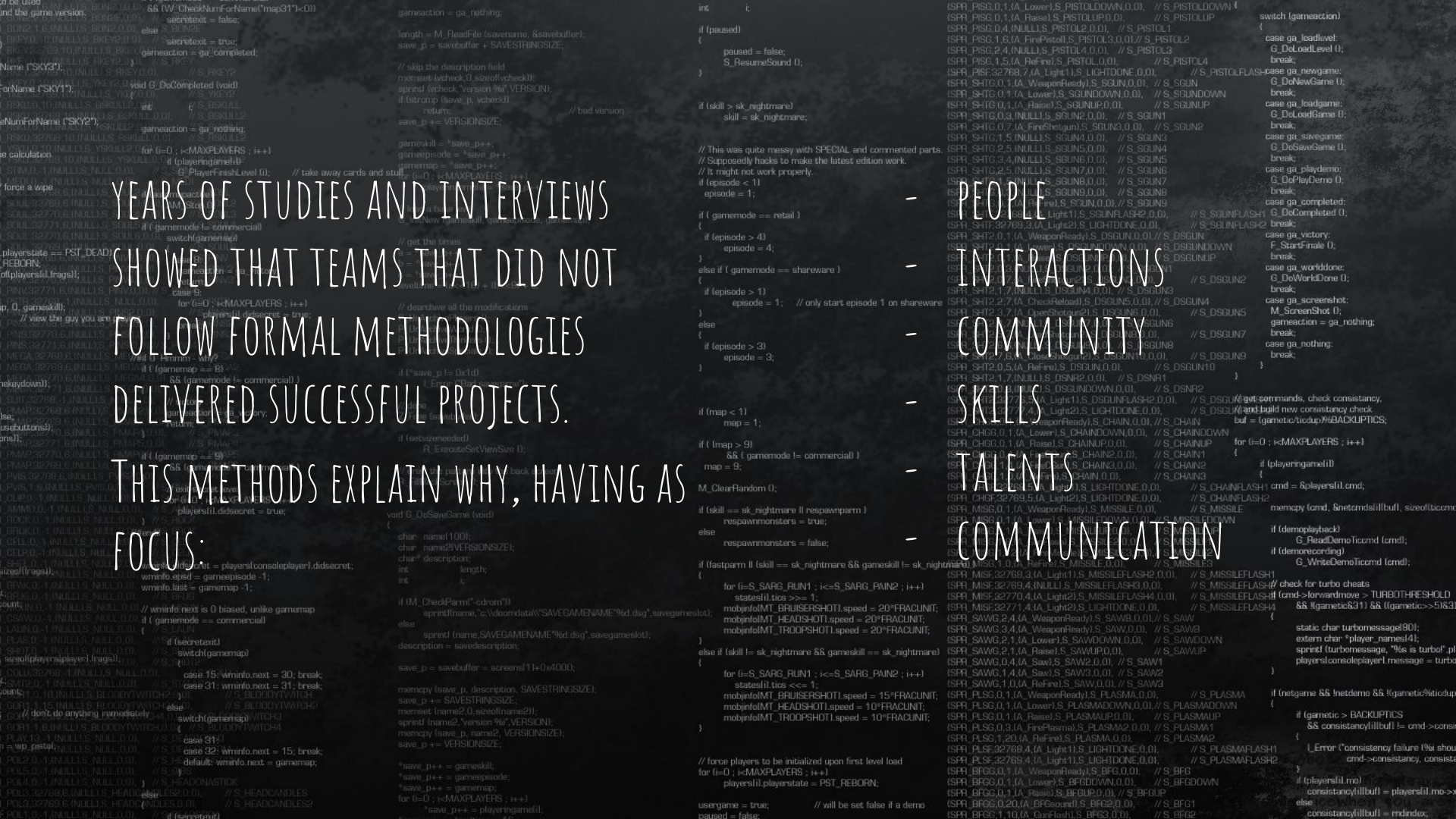
Nádia de Sousa Varela de Carvalho, up201208223

Lívia Carolinne Trindade de Lima Araújo, up201702858

António José Magalhães da Cruz Magalhães, up201404375

History

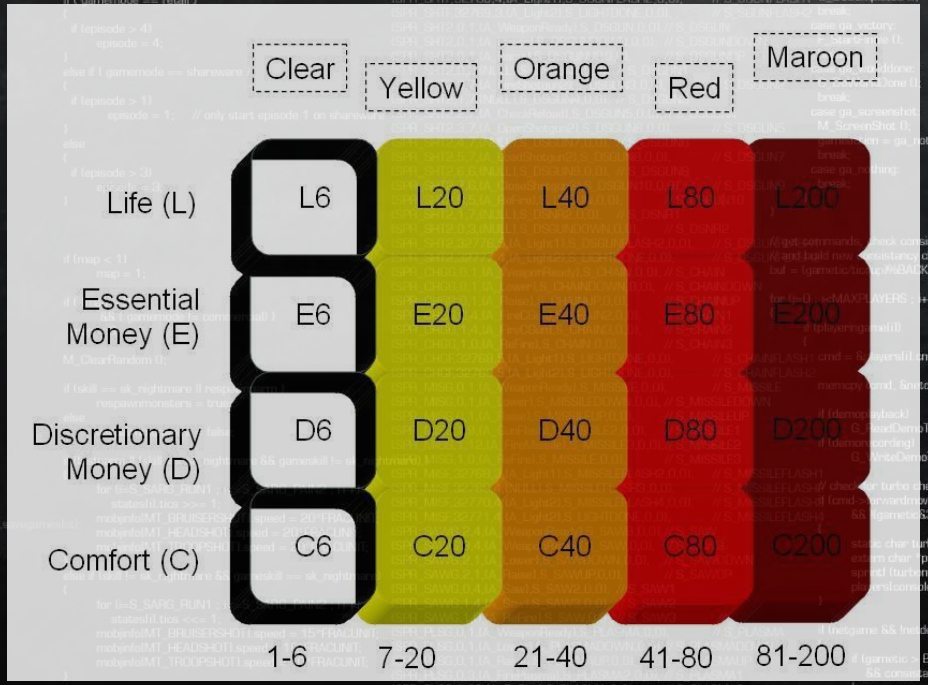
- Developed by Alistair Cockburn
- Mid-1990s
  - 1992: Start
  - 1997: named “Crystal”
  - 2001: Agile Manifesto
- It was one of the founding ‘Agile’ methodologies.


YEARS OF STUDIES AND INTERVIEWS
SHOWED THAT TEAMS THAT DID NOT
FOLLOW FORMAL METHODOLOGIES
DELIVERED SUCCESSFUL PROJECTS.
THIS METHODS EXPLAIN WHY, HAVING AS
FOCUS:
- PEOPLE
- INTERACTIONS
- COMMUNITY
- SKILLS
- TALENTS
- COMMUNICATION

Family of Methodologies

Organized by size and criticality:

- Crystal Clear
- Crystal Yellow
- Crystal Orange
- Crystal Orange Web
- Crystal Red
- Crystal Maroon
- Crystal Diamond
- Crystal Sapphire



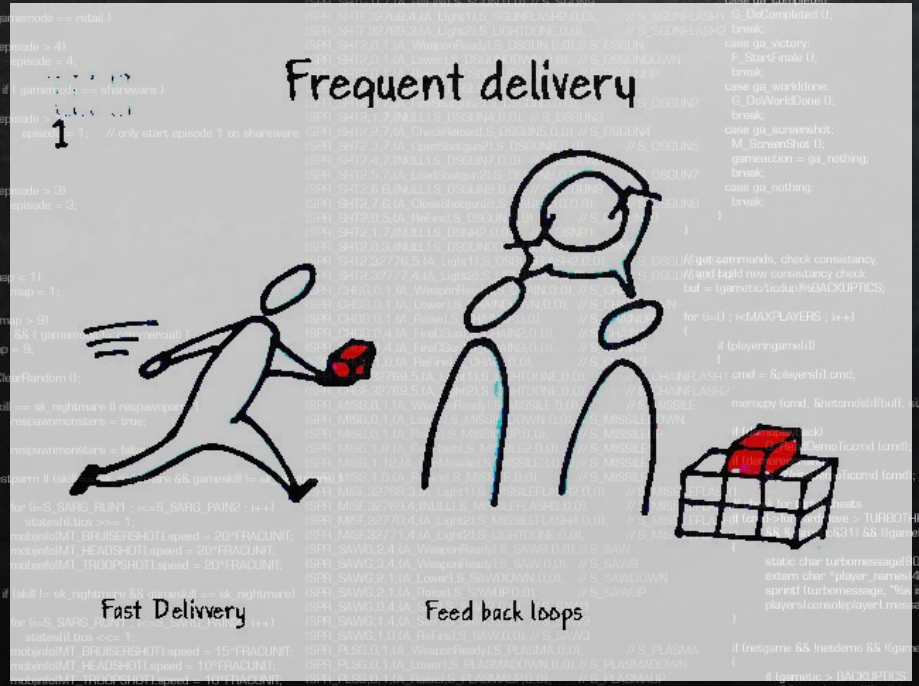
- Methodology - set of elements (e.g. practices, tools)
- Techniques - skill areas (e.g. developing use cases)
- Policies - dictate organizational musts

Proprieties

- Frequent Delivery
- Reflective Improvement
- Close or Osmotic Communication
- Personal Safety
- Focus
- Easy access to expert users
- Technical environment with automated tests, configuration management and frequent integration

FREQUENT DELIEVERY

- WEEKLY OR QUARTERLY
- MAXIMUM 4 MONTHS
- DECIDE FEATURES THEY MUST INCLUDE IN EACH RELEASE
- TEST AND USE OF FUNCTIONALITY AFTER EACH ITERATION



Reflective Improvement

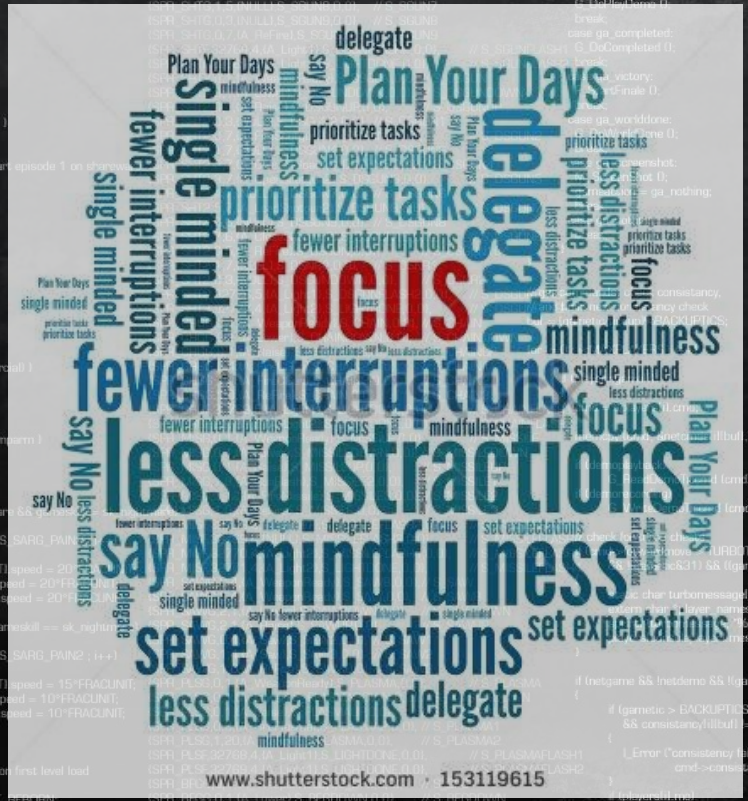
- Taking a break from regular development to find better ways for processes
- Meeting every couple of weeks to discuss what can be modified in the project

Close or Osmotic Communication and Personal Safety

- Team being together in a room and getting information to flow around it
- Questions that arise from the work can be rapidly answered, reducing the risk of errors
- People working on the project must be able to trust each other and feel free to speak up about issues

Focus

- Individual task
  - Two-hour-free of interruptions period for each developer
- Direction the project is heading
  - Each developer is assigned to a functionality for two days before being switched to another one



EASY ACCESS TO EXPERT USERS

- WORKING WITH A PERSON OF EXPERTISE:
  - ACTUAL/REAL-LIFE USER
- EXPERT ANSWERS ANY QUESTIONS

TECHNICAL ENVIRONMENT WITH AUTOMATED TESTS, CONFIGURATION MANAGEMENT AND FREQUENT INTEGRATION

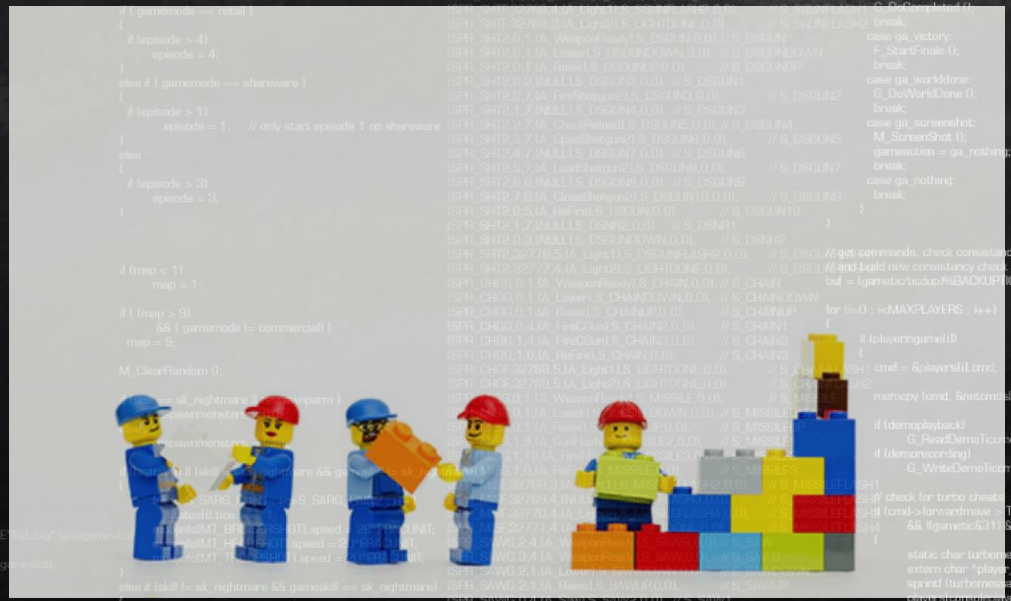
- CONTINUOUS INTEGRATION AND TESTING
- ERRORS, BREAKAGES, ETC CAN BE SPOTTED
- CHECKING-IN CODE INTO A REPOSITORY
- REMOVE IT BY REVERTING BACK OR UPDATING WITH CORRECT CODE

ROLES IN THE TEAM

- **Project Sponsor**: finance the project, deliver the mission statement
- **Senior Designer**: maintains the team structure, implements methodology and designs the system
- **Designers/Programmers**: creating screen drafts, design sketches and notes, common object models, source code, packaged system, migration code, and test cases
- **Business Class Designers**
- **Software “Documenters”**
- **Unit Testers**
- **Programmers**
- **Users**: help with use case and screen drafts

Strategies

- Exploratory 360°
- Early victory
- Walking skeleton
- Incremental
- Re-architecture
- Information radiators



Techniques

- Methodology shaping
- Reflection workshop
- Blitz planning
- Delphi estimation using expertise ranking
- Daily stand-up meetings
- Essential interaction design
- Process miniature
- Side-by-side programming
- Burn charts