PROJET DE DÉVELOPPEMENT COLLABORATIF (PDC)

INTRODUCTION AND ORGANIZATION

MASTER 1 ICE, 2017-2018

Available at https://combemale.github.io/teaching/m1ice/

BENOIT COMBEMALE
PROFESSOR, UNIV. TOULOUSE, FRANCE

HTTP://COMBEMALE.FR
BENOIT.COMBEMALE@IRIT.FR
@BCOMBEMALE

U.E. 803 - code MI0A803V - 6 ECTS – 50h
Libellé : « Projet de Développement Collaboratif (S8) » (PDC)

Coordinateur pédagogique	**B. Combemale**				
Modules prérequis	CPOA, EC, GL, MP1				
Modules liés	IPC				
Compétences visées	• S'approprier les techniques abordées antérieurement : conception et programmation objet (UML, java), développement dirigée par les modèles, patrons d'architecture et de conception, refactoring, tests, analyses statiques... • S'approprier les méthodes de développement industrielles abordées antérieurement (cycle de vie incrémental, méthodes et principes agiles, DevOps) • Maitriser certains outils de développements et de collaboration (ex. Eclipse, Maven, SVN/Git, Github, JENKINS, …) • Savoir mettre en œuvre des techniques et des outils de gestion de projet (incl. gestion des exigences et maîtrise des changements, et traçabilité)				
Contenu	Heures	Cours	TD/TP	Modalités de contrôle des connaissances	Intervenants
Projet collaboratif long se déroulant selon 3 phases, en respectant une méthode incrémentale et les principes agiles :				1 note finale (0,5 * note individuelle + 0,5 * note de groupe)	B. Combemale Vacataires (F. Coulon)
PDC a – Lancement du projet	18	6	12		
PDC b – Elaboration et production d'un 1er prototype	14	6	8		
PDC c – Production, validation et recette du produit final	18	6	12		
Remarques :	RAS				

Objectives

- Technical skills
  - Development, with state-of-the-art design and architectural choices
  - Modern application domains and frameworks,
  - Set up an industrial and collaborative development environment
  - Application of agile principles
  - Application of modern and industrial principles for releasing the solution

  ⁉️ *You should apply your skills from V&V, DevOps, IM, (Stats?)…*

- Soft skills
  - Project-team organization and management
  - Communication and valorization
  - Evolve in an international context, including the use of English for communication materials, project management and project implementation (specification, code, etc.)

General organization

- Workgroup sessions for project implementation
  - Technical support
  - *Un*scheduled lectures (opportunistic mode) on emerging topics
- Project meetups
- Final presentation

- Project team:
  - Composed of 5-6 complementary members (fill-in: https://goo.gl/jGjGA3)

- Stakeholders:
  - Team members: M1ICE students (all timeslots)
  - Product owner: Benoit Combemale (project meetups)
  - Technical support / opportunistic lectures: Benoit Combemale & Fabien Coulon (all timeslots)

Project

Tool-supported environment for domain-specific data querying, analysis and monitoring

- DSL for data querying, analysis and visualization
- Front-end for monitoring (real-time, and possibly interactive), and possibly prediction (interpolation, ML)

Think about a ***domain-specific*** Shiny, cf. http://shiny.rstudio.com ;
or Eclipse ICE, cf. https://www.eclipse.org/ice

1 Project, N instances!

- DSL
  - Expressiveness:
    - *Querying*: which abstractions? which time/space management?
    - *Analysis*: which paradigm? IFTTT, ECA, state-based, imperative, rule-based…
    - *Visualization*: binding to plotting? domain-specific widget?
  - Technical stack? Web, Eclipse-based, etc. external/internal DSL
  - Deployment? Docker?
- Frontend (interactive?)

Example of (real-time) open data

- DATA.toulouse-metropole: https://data.toulouse-metropole.fr (e.g., Real Time TISSEO API)
- Twitter: https://developer.twitter.com
- Flight tracking (e.g., https://opensky-network.org)
- https://data.worldbank.org
- https://openweathermap.org/api
- https://github.com/abhishekbanthia/Public-APIs
- https://any-api.com/
- https://www.programmableweb.com/apis/directory
- https://github.com/toddmotto/public-apis
- Etc.

Technologies (examples, strict minima)

- Eclipse/EMF/Xtext/Sirius, EMFRest, DSLForge, …
- Java/Xtend, ATL, Acceleo, …
- Angular, …
- Github, Travis, Sonar, Docker…

Timeline

- **W5-6-7, 2018** ***(P3: take-off)*:**
  - Project presentation and definition of the teams: Jan. 30th, 8h30-10h30
  - **Sprint #0** (W5, 4h +4): training, set up of the development/project infrastructures, main design choices, backlog definition
  - **Sprint #1** (W6-7, 12h +5): first running prototype, all main scenarios should have the nominal functional chains (i.e., stories)
    - *project meetup: Tuesday, February 13th, 2018, 14-16h*
- **W13-14-15, 2018** ***(P4: flight plan)*:**
  - **Sprint #2** (W13-14-15, 14h +5): running prototype according to the selected stories
- **W22-25, 2018** ***(P5: landing)*:**
  - **Sprint #3** (W22-23-24, 14h +5): intermediate delivery, including customer release (as defined with the product owner)
  - **Final presentation and awards** (W25, 4h +4): 30min per project, including presentation, demo and Q&A
    - *3h: project presentations*
    - *1h: deliberation, awards and debrief with students.*

✓ Project meetup = debrief of the completed sprint, backlog grooming, task board definition for next sprint.

Great team force, but tight schedule . . .

Optimize your teamwork!

COACHING
GOAL
TEAM
BUILDING
TRAINING
SOLUTION
100 %
COLLABORATION
MOTIVATION

Guidelines

- Set up a shared **development environment**, incl.:
  - IDE (design and implementation), and the underlying technological stack (e.g., exec. platform, middleware…)
  - Shared code repository (Github) + collab. tools (issues, trackers, tasks, release…)
  - Complete CI (e.g. Travis), incl. build, test, deployment, documentation
- Provide a project **management platform**, incl.:
  - User stories
  - Task board
  - Planning and tasks management
- Set up the appropriate **communication channels** for your development team, and the relation with the product owner (mailing list, slack, …)
- All **developments** must include:
  - Design: functional, structural, behavioral and implementation modeling
  - Implementation: source code and project descriptors
  - Test: unit, integration and system tests
  - Documentation (user/developer)

- Use English, provide polished documents, clean source code, etc.
- Be formal, professional, pro-active, pragmatic, solution-driven, and keep going!

Evaluation

Explicit KPIs must be defined during kickoff in collaboration with the product owner

- **Technical delivery *(coef. 0.5)***
  - Attributed by the product owner
  - Evaluation criteria:
    - Overall relationship between the project team and the product owner
    - Quality of the final delivered solution
      ⁉ The product owner may require specific outcomes in addition to the code, such as an annotated screencast of a demonstration, a written tutorial of the solution, etc.
    - Documentation, test

- **Overall project implementation *(coef. 0.5)***
  - Attributed by the jury of the final presentation, composed of B. Combemale, F. Coulon
  - Evaluation criteria:
    - Project management: set up, implementation, delivery
    - Development environment: right tools for right tasks
    - Conformity to the project guidelines
    - Overall quality of the developments
    - Overall quality of the presentation of the project, and description of the solution

Enjoy, and keep going!

be pro-active, and make it Your project!