

PROJECT MANAGEMENT

Part 1 : An energized and organized team



Tianxiao LIU - Master IISC 1 – CY Cergy Paris Université

TRUST

- We work together effectively and **without fear**.
- **Four stages** : Forming → Storming → Norming → Performing
 - This won't magically appear !
- Team members : think of the rest as "us"
 - To notice things that need to be done.
- Taking time to help others won't make you look unproductive.
 - Be treated with respect when asking help or disagreeing with someone
- **We need strategies to gain trust for each other !**

A TYPICAL EXAMPLE OF FEAR

- What's the problem ?
 - Programmers : no respect for the testers' abilities
 - Testers : mission : shooting down the programmers' work !
- **No “us versus them” ...**
- Solutions
 - Testing takes skill and careful work, as programming
 - Focus on team's joint goal : releasing a great product
 - Finding a mistake is not an occasion for celebration or gloating...
 - Mistakes aren't a sign of incompetence or laziness...
- Sit together → Sharing thoughts breaks down barriers !

WE NEED SOME HUSTLE

“Things may come to those who wait, but only the things left by those who hustle”. --- A. Lincoln

- Hustle = Energized, productive work
- Stakeholders are often “war wounded soldiers”
 - Slipped schedules, unfixed defects and wasted money
 - They are put in the uncomfortable position of relying on your work...
 - Your team consumes thousands of euros every week in salary and support
- Think of it in their position...
- Two kinds of results will speak clearly
 - **Working product and delivering on commitments**

EARNING TRUST BY MANAGING PROBLEMS

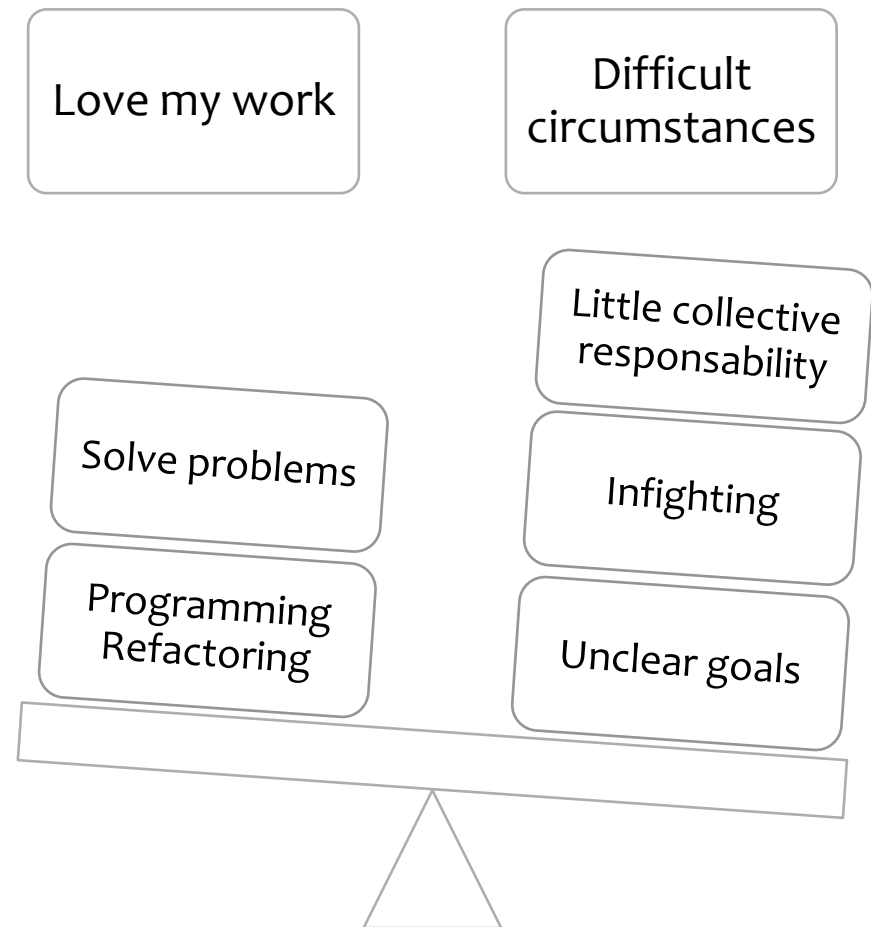
- Limit your exposure to problems.
 - Work on the hardest, most uncertain tasks early in the iteration.
- The bigger the problem, the sooner you should disclose it.
 - Addressing a problem successfully can build trust like nothing else.
- Relying on overtime indicates a systemic problem.
 - That saps the team's energy and reduces our ability to absorb problems.
- **BUT**, in the context of academic projects, that often means you are lack of required skills or experiences...
 - Do you really have a choice ?

A TRUE STORY ABOUT TRUTH-TELLING

- **If you are sure that you can't make the deadline work, are you going to tell your customer the TRUTH ?**
- A true story of an interviewed project coach
- Key vocabulary and information for understanding the story
 - **Story** = User story
 - A quick description of what a user needs to do with the system
 - Forbade (forbid), this action was the big mistake.
 - Brooks' Law : *Adding manpower to a late project makes it later.*
 - **No miraculous turnarounds**
- **What have you learned from the story ?**

ENERGIZED WORK

- We work at a pace that allows us to do our best, most productive work indefinitely

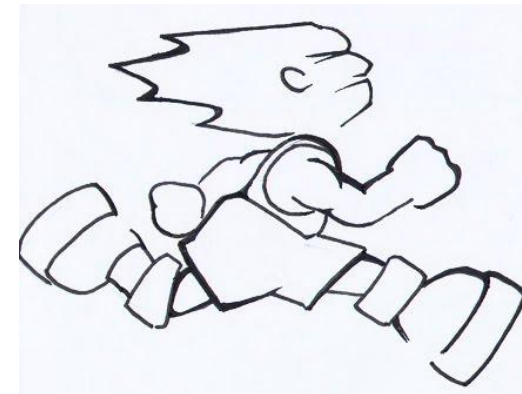
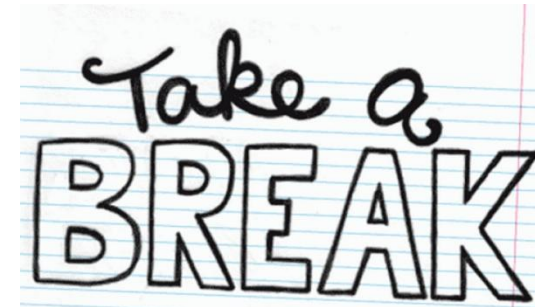


SUPPORTING ENERGIZED WORK

- It isn't easy : no way to force someone to be energized
- Pair programming encourage it : tired but satisfied
- Having health food available in the workspace
- We are motivated to do good, intellectually challenging work.

THE ART OF TAKING BREAKS

- Stay home when you are sick.
 - You risk getting other people sick too.
- Stop and take a break
 - When you are making more mistakes than progress
 - Angry concentration, cursing at the computer, abrupt movements
 - Going dark without talking in a highly collaborative environment
- About the sprints
 - Sprinting to finish the line is one thing, sprinting for miles is another.



INFORMATIVE WORKSPACE

- We are **tuned in** to the status of our project.
- An informative workspace broadcasts information into the room. → It helps at breaks.
- Objective : We know the state of your project without interrupting team members.



A healthy project

Not tension,

but activity

VS

An unhealthy project

Quiet

and tense

BIG VISIBLE CHARTS

- Big visible charts can help a lot to have an informative space.
- Useful examples
 - Iteration or release planning
 - Team calendar : dates, numbers...
 - Pairing combinations

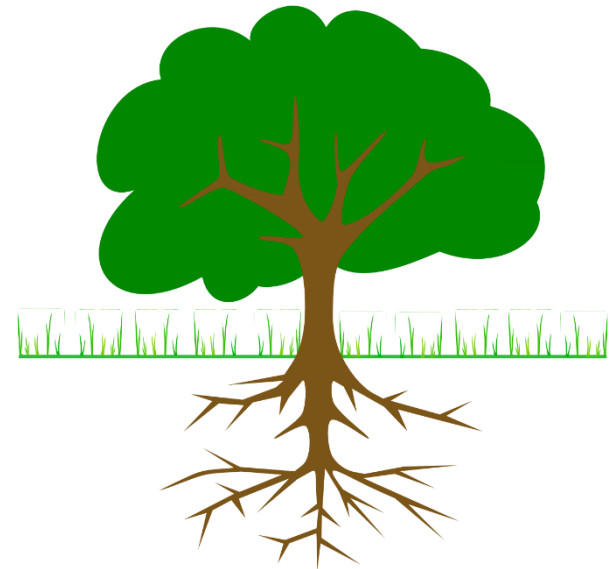
Pairing	Luc	Catherine	Jean	Paul
Paul	GUI design			
Jean		Testing		
Catherine	Code refactoring			
Luc				

KEY POINTS FOR YOUR WORKSPACE

- Simply poking your head into a project room should give you information about the project !
 - You can never have too many white boards.
- Don't rush to computerize everything.
 - "Do Not Erase" label for more than a few days may indicate a problem.
- We need to do **improvements charts**.
 - Never use workspace charts in a performance evaluation.
- Project manager : **If people won't take responsibility, perhaps you are being too controlling.**

ROOT-CAUSE ANALYSIS

- **We prevent mistakes by fixing our process.**
- **Murphy's Law** : *If something can go wrong, it will.*
- Don't blame someone for screwing up.
- What is it about the way we work that allowed this mistake to happen ?
- How can we change the way we work so that **it's harder for something to go wrong ?**
- Classic approach : **ask "why" five times.**



FIND AND FIX ROOT-CAUSE

- An example for understanding "five times question" approach
 - **Problem** : A lot of time spent getting the code into a working state, when we start the work of a new task.
 - Why ? (1st time) : build often broken
 - Why ? (2nd time) : people check in code without running the tests
 - Here, it's easy to stop and say : We found the problem ! → Dig deeper !
 - Why ? (3rd time) : tests take longer to run than people's availability
 - Why ? (4th time) : tests spend a lot of time in database setup and teardown
 - Why ? (5th time) : **because our design makes it difficult to test business logic without touching the database**
- Fixing root-cause requires the cooperation of the people involved
 - In team, facilities, organization...

RETROSPECTIVE

- We continually improve **our work habits**.
- No process is perfect and our team is unique, as are the situations.
- Retrospectives : great tool for updating process to match situations.
- When do we need it ? Release, iteration, daily, weekly...

Norm Kerth's Prime Directive (NKPD) :

Regardless of what we discover today, we understand and truly believe that everyone did the best job they could, given what they knew at the time, their skills and abilities, the resources available, and the situation at hand.

CONDUCTING AN ITERATION RETROSPECTIVE

- Brainstorming (everyone agrees with NKPD)
 - On the whiteboard, write down the events that fall into :

Enjoyable

Frustrating

Puzzling

- For each event, tell

Same

More

Less

- Two results of retrospective
 - Sharing ideas gives the team a chance to **grow closer**.
 - Coming up with a specific solution gives the team a chance to **improve**.