

collaborative development of interactive software systems teamwork

alexandre r.j. françois

visiting assistant professor of computer science
tufts university

research assistant professor of computer science
viterbi school of engineering, university of southern california



outline

project organization

team organization

tools

project

challenges

- complex – more than just writing a few functions
- short time – work collaboratively
- need to organize!

questions:

- what is the project goal?
- what is success? metrics?

software engineering

“technique is a process of refinement that characterizes formalized knowledge.” [FS04] p.6

not just common sense!

continuous iterative improvements

design (in software context)

“do things intentionally”

project

consider:

scope

architecture

planning

testing

implementation

how:

answering risk with “design”

game design vs. software design?

architecture, code, etc.

development and testing

documentation

team responsibilities

typical team work in game development: [FS04] pp.546-547

prepare development plan

gather technical requirements for game

=> s/w requirements document

design software system that supports the game features

=> s/w design description

construct the software

=> code documentation

prepare and execute test plan

prepare and execute deployment plan

caveats

assumes game design already completed

linear process!!

team organization

who does what?

collaborative teams (concurrent vs. parallel)
fluid membership (?)

communication and documentation are key!

management vs. leadership [TP#911]

possible distinction:

management = do things right

leadership = do the right thing

again, not just common sense!

communication pathways

change proposal and control protocols

design documents

requirements document

software architecture

development plan

milestone schedules

top 10 risk list

more: [FS04] pp.557-559

communication tools

collaborative authorship

GoogleDocs

Wiki

version control systems (code)

CVS

Subversion

integrated tools

sourceforge.net

architectural framework / design language

sai

reasons teams fail

methodology shortcuts
worthless methodologies
loss of scope
unrealistic schedule
poor estimations
too much enthusiasm

more: [FS04] p.572