Capstone Project: From Software Engineering to “Informatics”

Hadar Ziv, Sameer Patil
Department of Informatics
University of California, Irvine
ziv@ics.uci.edu
UC Irvine / Orange County, CA

Irvine/Orange County close to Hollywood/Los Angeles, but we have our own identity…
UCI / ICS / Project Course

• (formerly) Department of ICS
  – Established 1968 (recently celebrated our 40th)
  – Undergraduate and graduate-level degrees

• Project course in software design
  – Since the 1980s
  – Small group project
    • 4-5 students per team
    • Single quarter (10 weeks)
  – Project only
    • Everything was taught/should have been learned prior
  – External client
    • Must be “real” project, “real” requirements, “real” client needs
Project Course Objectives

• Hands-on experience
  – Learning by doing!

• Real-world experience
  – “real” projects of value to “real” customers
  – Customers receive not only software, but a complete set of SE deliverables
  – Software is either deployable, prototype or proof-of-concept (“trial run”)
    • UCI University Club  http://www.uclub.uci.edu/

• Technical skills / SE skills
  – Requirements elicitation, use cases, user stories
  – HCI, UI design, web design
  – Software architecture, OOAD/UML, OO programming
  – Lifecycle models, incremental and iterative, Agile/SCRUM

• Non-technical / people skills
  – Project management, team / time / resource management skills
  – Communication / collaboration / coordination skills
School of ICS / Informatics

• Donald Bren School of ICS
    • Opened/Moved to Bren Hall 2007
  – Only CS school in University of California
    • 1,000 undergraduates / 300 graduate students
    • Several departments, including Informatics

– Informatics Faculty with degrees/background in
  • HCI/HCC/CSCW
  • Psychology/cog sci
  • Ethnography
  • Anthropology...
Informatics Curriculum

– Broader curriculum covering topics such as
  • HCI / user-centered design / UI and web design
  • Computer supported collaborative work
  • Social analysis / Social aspects of computing
  • Project management
  • Usability engineering
  • Information visualization, Information/data mining, ubiquitous computing...

– Extend the project course to three quarters
  • To include and require application of such topics
  • Required course in the final year of study
  • Core requirement towards earning an Informatics degree
Informatics Capstone Project

• We’ve been teaching the year-long project
  – Here is what people are saying…

• We’ve observed three kinds of changes with the increase in project duration
  – Some aspects increased proportionally
  – Some remained constant or attenuated
  – Some increased disproportionally
Equi-Proportional “Growth”

• Project scope
  – Scope / Size / Complexity are proportionally larger
    • E.g., web systems

• Software requirements
  – The number / extent / complexity of use cases / user stories
    • E.g., healthcare-system requirements

• Software development
  – Design / implementation / deployment effort

• Grading / Assessment
  – Project-based
    • Project-based “events”, such as document-delivery milestones
    • Team-based “events”, such as live demos in class and to customers
  – Individual-based
Constant / Little or No Growth

• Infrastructure setup
  – Hardware, web space, web servers, etc.
  – Software tools, web tools, PM/time-tracking tools, etc.

• Team building
  – Getting to know each other, their users, customers, stakeholders
  – Team room, project wall, etc.

• Learning curve / Time and effort
  – Programming languages, software tools, web tools, PM/time-tracking tools, etc.
Disproportionally Large Growth

• In the single-quarter project,
  – Little time for serious usability engineering
  – Straightforward project management
  – Little time for serious testing / quality readiness

• In the year-long project,
  – Usability planning, iterative testing, incorporating test results and feedback
  – Project planning, scheduling, execution and monitoring
  – Testing at multiple levels, regression testing, release planning, multiple builds, releases, production-quality
What Works Well

• We continue to be pleasantly surprised that
  – HCI/HCC works
    • Students and customers took to human-centered design
    • Students promoting HCC in their jobs/internships
  – SE works
    • Iterative/incremental/agile/test-driven development
    • Students delivering complex software of high quality
  – Team-based, active learning works
    • Students learning within their teams, peer reviews across teams, and active learning in the classroom
What Doesn’t Work so Well

• We continue to face challenges of
  – Senioritis
  – Students who do less, contribute less, and ride on the coattails of others
  – Students struggle to maintain focus/motivation and creativity/productivity over three quarters
  – Customers struggle to maintain focus/motivation over three quarters
Final Remarks

• The students are learning by doing
  – Real users with real needs
  – Real requirements from real customers
  – Apply all elements of prior Informatics curriculum
  – Iterative and incremental development
  – Peer reviews, active-learning in the classroom

• We are learning by doing
  – Actively teaching the course
  – Iterative and incremental course development
  – Continuous feedback / reflection / improvement

• If you want to start a Capstone project
  – Good papers in this session and in many other sources
  – Talk to us / We can help / We can learn from each other!
Thank you!

• Questions?
What is Informatics

• Overheard at CSEET 2010…
  – Computing is not just about code (Lynn Carter, 03/10/2010)
  – Need solutions that actually help users, work for users (Lynn Carter, 03/10/2010)
  – Human aspects such as usability are important (Mary Shaw, 03/10/2010)