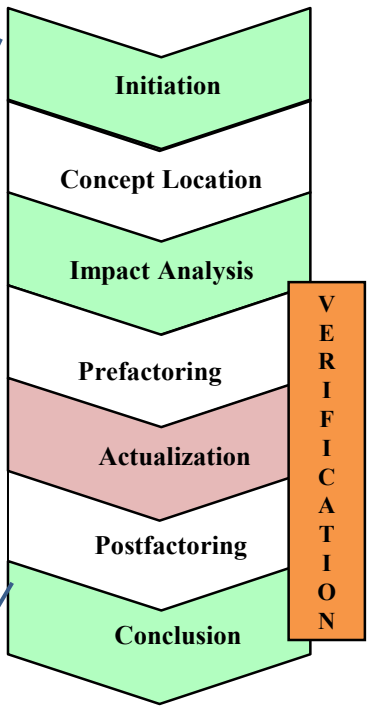
Which Practices are Suitable for an Academic Software Project?

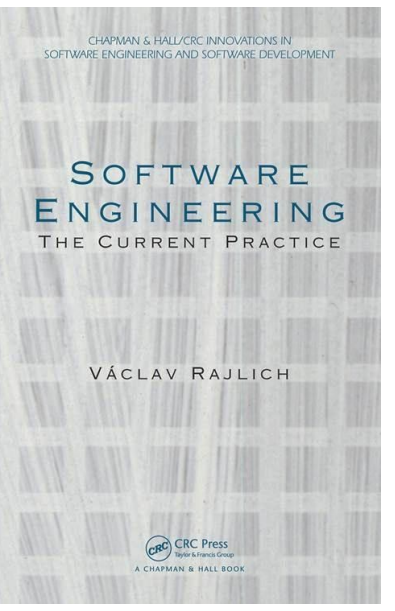
Václav Rajlich, Jing Hua, Wayne State University

rajlich@wayne.edu

Organizational vs. code development practices.

The code development practices classified by phases of Phased Model of Software Change (PMSC)





Data: The practices of three academic projects:

	feniCS	Dalton	BrainSpace
web site	http://fenicsproject.org/	http://daltonprogram.org/	http://www.cs.wayne.edu/~jinghua/NSF/ImagingInformatics.htm
# contributors	7 institutions + open source participants	more than 60, but few people involved in any particular time	~10 contributors in the history of the project
languages	C++, Python	Fortran 77/90, C, C++	C++
availability	open source	free, licensed	available to collaborating users
domain	differential equations	molecular chemistry	visual analytics
		Organization	
roles	equal peers, core team, participants choose how much effort, dedicated tester	board, lab supervisors, students	supervisor (varying availability), project leader, advanced students (research assistants 50% involvement), beginners
coordination, monitoring	LaunchPad tool, distributed team	ad hoc, occasional meetings, distributed team	weekly meeting during academic year, monitoring by project leader
domain knowledge	developers are domain experts	developers are domain experts, users also provide expertise	developers are domain experts, users (physicians) also provide expertise
		Code development	
initiation	exploratory specifications, personal initiative	exploratory specifications, participating lab priorities	both defined and exploratory, selection based on funding and stakeholder needs
concept location		solved by code ownership	solved by code ownership
impact analysis			estimated by supervisor and project leader
actualization	undisclosed	ad hoc	ad hoc
refactoring	yes	no	very few
verification	inspections, regression tests	regression tests	functional tests, efficiency tests, inspections
conclusion	Buildbot used for build, test, and release	four releases since 1997	irregular iterations, based on funding and academic schedule, releases about 6 months
Open problems	coordination of different requirements related to the same functionality	hard to integrate branches, errors in the underlying theory	management of variants of the main software

Academic managers and developers are often specialist in fields other than software engineering.

These studies may help them to select the appropriate practices for their projects.