Overview of the Class Syllabus for SWE 625

- Why Take SWE 625?
- Scope
- Motivation
- Biography
- Administration
- Course Text
- Major Topics
- Course Background Requirements
- Course Evaluation Procedure
- Lecture Topics
- Blackboard Learn – course materials and online presentations
Why Take SWE 625?

• Successfully managing software intensive projects is a **priority** for the industrial, government and academic organizations

• The ubiquity of software and its critical role require fundamental shifts in software engineering management and engineering to maintain competitive advantage

• The course helps participants to **rapidly deploy innovation with confidence** within this shifting landscape by:
  • Applying new principles in software engineering management for software intensive systems
  • Developing new practices for **enabling business/mission capability with software innovation**

• **Equips students** in applying new management techniques in today’s competitive job market
Scope of Software Engineering Management

Mission Focused

- System of Systems – all types
- Networked Hardware/Platforms
- Infrastructure
- Applications
- Workforce: People who digitally connect to cyberspace

“Software is the building material for modern society”

Source: SEI
## Fundamental Shifts in Software Management & Engineering

As software and systems are increasingly becoming **bigger**, **more complex**, and **intertwined**, software management and engineering and the roles people play are evolving in response.

<table>
<thead>
<tr>
<th>Time</th>
<th>Developers write code</th>
<th>Models generate code</th>
<th>AI/ML assists in generating models/code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software release based on milestones (typically 12 – 24 months)</td>
<td>Continuous integration and continuous deployment (CI/CD)</td>
<td>Automated release-observe-refine</td>
<td></td>
</tr>
<tr>
<td>Collect data and evidence from past projects to make predictions</td>
<td>Moving beyond prediction to determining causality</td>
<td>Feedback of data and results to re-train models</td>
<td></td>
</tr>
<tr>
<td>Software and hardware must work together</td>
<td>Increasing diversity of languages, platforms, hardware &amp; systems must be made to work together</td>
<td>Systems of people, policies, sensors, software, hardware, etc., continuously learn ways to work together</td>
<td></td>
</tr>
<tr>
<td>Developers do nearly everything</td>
<td>Developers determine processes and rules and create automation</td>
<td>Machines continually learn what to do to achieve goals</td>
<td></td>
</tr>
<tr>
<td>Black box test for correctness</td>
<td>Formal analysis of correctness</td>
<td>Mathematically verified enforcers watch rest of system</td>
<td></td>
</tr>
<tr>
<td>Human in the loop (humans invoke computers)</td>
<td>Humans on the loop (humans monitor computers)</td>
<td>Humans out of the loop (computers notify humans only when needed)</td>
<td></td>
</tr>
</tbody>
</table>
Software is ubiquitous and U.S. national security relies on software. Well-equipped and well-trained warfighters provide the capability necessary to defend the nation, but software critically enables that mission. The ability to develop, procure, assure, and deploy software is central to national defense and integrating with allies and partners.

Speed and cycle time are the most effective metrics for software. Software is a critical element of the Department’s approach to executing missions, collaborating with allies, and managing its operations. DoD needs to deploy & update software at the speed of (mission) need and execute within the OODA loop of our adversaries to maintain advantage.

The OODA loop is the cycle observe–orient–decide–act, developed by military strategist and United States Air Force Colonel John Boyd. Boyd
Conclusions - Defense Innovation Board Software Acquisition and Practices (SWAP) Study – 11 Jan 2019

• Software is made by people, for people, so digital talent matters. DoD’s (and commercial) current personnel processes and culture will not allow its military and civilian software capabilities to grow nearly enough. New mechanisms are needed for attracting, educating, retaining, and promoting digital talent, and providing the ecosystem that enables them to succeed.

• Software is different than hardware (and not all software is the same). Hardware can be developed, procured, and maintained. Software is an enduring and evolving capability that must be supported and continuously improved throughout its lifecycle. The DoD (and Commercial) acquisition process and culture need to be streamlined for effective delivery and oversight of multiple types of software-enabled systems, at scale, and at the speed of relevance.
Biography

DR. KENNETH E. NIDIFFER, PMP
Director of Strategic Plans for Government Programs
Software Engineering Institute, Carnegie Mellon University

Dr. Nidiffer has over fifty-seven years of experience in the marketing, research, development, support, maintenance, and acquisition of software-intensive systems. His 24-year career in the U.S. Air Force (where he retired as a full colonel) is marked by several firsts in the area of software implementations, such as, first space-based compiler, first command-hardware in the loop simulation, a series of development/process standards, etc. From 1983-1986 he helped establish several noteworthy contributions, such as, the Software Productivity Consortium; the Software Project Management Program at the Defense Systems Management College; the George Mason Software Engineering Program and the Software Engineering Institute. At the Software Productivity Consortium, he launched the Consortium’s business initiative in software process improvement, which became one of the largest programs in the world.

In 1991, Dr. Nidiffer left the Consortium to serve one of its founding members, Northrop Grumman, as Director of Systems Design and Development, Data Systems Division, and then as Director of Technical Operations, External Data Systems division, where he directed over 500 engineers and support personnel in the successful development of a variety of C4I, MIS/logistics, and high-speed computing applications.
Biography

In 1995, he joined Fidelity Investments Systems Company as Senior Vice President of Quality and Systems Assurance to lead a team of 165 professionals in implementing Total Quality Management, best-in-class software engineering processes, and the largest financial services test environment. He rejoined the Consortium in 1997 as Vice President for Business Development growing the membership from 50 to 100 members. In 2007 he joined the Software Engineering Institute and served as a principal senior engineer to focus on promoting key software engineering technologies that support government programs and retired in 2020.

Dr. Nidiffer has been widely published in the systems and software engineering community. He received his B.S. degree in Chemical Engineering in 1962 from Purdue University, Indiana, a M.S. degree in Astronautical Engineering in 1969 from the Air Force Institute of Technology, Ohio, a MBA degree from Auburn University, Alabama in 1975 and his D.Sc. degree from George Washington University, Washington D.C. in 1988.

He is a member of the Program Management Institute (PMI); the International Council on Systems Engineering (INCOSE); the Air Force Association (AFA); Senior Member of the Institute of Electrical and Electronics Engineers (IEEE) and Member of the IEEE Professional and Activities Board (PAB); the Inter-National Committee for Information Technology Standards (INCITS)/Software and Systems Engineering (INCITS/SSE) Technical Committee, Senior Member of the American Institute of Aeronautics and Astronautics (AIAA); member of the National Defense Industrial Association (NDIA Systems Engineering Division); Chair of the NDIA/OSD (DDR&E) Industrial Software Committee and Co-Chair of the NDIA Systems Engineering Education and Training Committee.
Biography

Ken is a certified logistician; a Professor Emeritus of the Defense Systems Management College; Industry Advisor on George Mason’s Computer Science Education Committee; a Project Management Professional; and an adjunct engineering professor in graduate engineering at George Mason University for over 28 years.

Dr. Nidiffer is a man of faith and a family-oriented person. He has been married for 57 years to the former Mary Emma Walsh of Havana, Florida and they have three daughters: Sheri, Kristi and Kathi and three grandchildren. In 2002 and in 2007, he was selected as the School of Information Technology’s adjunct professor of the year in Software Engineering and received special recognitions for his GMU adjunct teaching service in 2009, 2013, 2017, 2018, 2019 and 2020.
• COURSE TITLE: Software Engineering Project Management (SWE 625)
• INSTRUCTOR: Professor Kenneth E. Nidiffer
• SEMESTER CLASSES: 25 Jan to 3 May 2021, including turning-in final exam
• SEMESTER FINAL EXAM: Take Home
• CLASS TIME/METHOD: 1920 – 2200; on-line synchronous
Meeting Arrangement Mechanisms:
- Establish a ZOOM meeting
- By appointment in on-line class session
- By the Internet – knidiffe@gmu.edu
- By setting-up a conference call
- By setting-up a video-teleconference (VTC)
- Department Administration Assistant
  - Ms. Michele L. Pieper: 703-993-1530
CONTACT INFORMATION:
- Internet/E-mail: knidiffe@gmu.edu – Best Method Overall
- Oral Communication Mechanisms:
  - Method 1: (703) 217-0215 (Cell Phone) or Text Message – Good Method
  - Method 2: (703) 455-4021 (Home Phone Number) - Good Alternative Method
TEXTS*:
Textbook No. 1: Title - Managing and Leading Software Projects
Author of Texts: Dr. Richard E. (Dick) Fairley
Publisher: John Wiley & Sons, Inc.
Options to Obtain:
1. Can Pick-up at University Bookstore (located in the George W. Johnson Center)
2. Order on-line
3. Obtained previously owned book

* Students are expected to study and understand the contents of the course text books
COURSE OVERVIEW
SOFTWARE ENGINEERING PROJECT MANAGEMENT 625

COURSE PREREQUISITES:
Undergraduate courses or equivalent knowledge in structured programming in a high-level language, data structures, discrete mathematics, and machine organization or assembly programming.

COURSE DESCRIPTION:
This course is concerned with processes involved in project planning; organizing; staffing; estimating; measuring and controlling; communication, coordination and leadership; and risk management. Topics covered include lifecycle delivery approaches; process and engineering product development models with special emphasis on the best practices contained in the Capability Maturity Model Integrated (CMMI©) constellations and product standards. The course also stresses the Program Management Institute’s Program Body of Knowledge (PMBOK©) and the Software Engineering Body of Knowledge (SWBOK).
COURSE OBJECTIVES:

Upon completion of this course, students will know how to develop a software project management plan for software intensive systems; how to set up monitoring and control mechanisms; how to allocate and reallocate project resources; how to track schedule, budget, quality, productivity, and progress; understand the CMMI© frameworks and how to plan for the installation and sustainment phase of the system life cycle. They will understand the importance of the work breakdown structure and its relationship to the delivery lifecycle, resource planning and execution, and progress and product measures from both a project and enterprise perspective. In addition, they will understand the relationships among quality assurance, configuration management, verification and validation, and test and evaluation. They will also gain an understanding of the key issues in costing and pricing units of effort, motivation of workers, agile development, Secure DevOps, leading project teams, machine learning, ethics and total quality management.
MAJOR TOPICS:

A taxonomy of management functions; corporate goals and objectives; system, project and product (functional and non-functional) requirements; architectural frameworks; best practice frameworks, such as the Adaptive Acquisition Framework (AAF), cost estimation techniques and models; software process development models with special emphasis on the CMMI® and software systems engineering delivery models; technical methods; documentation, quality assurance, configuration management, verification and validation, test and evaluation; staffing plans; monitoring and controlling mechanisms; standards (e.g. IEEE/EIA 12207 and IEEE Std. 16326™), policies and acquisition frameworks (i.e. Defense (e.g. DODI 5000.02, Defense Acquisition Guidebook (DAG) and Commercial (e.g. Infrastructure Service Provider (ISP) /Application Server Provider (ASP) frameworks; Platform as a Service (PaaS), Software as a Service (SaaS)), and procedures; work packages, schedules, budget, accounting systems, costing and pricing units of effort; risk management; post deployment software support; leadership, ethics, team building and total quality. Also, Defense Innovation Board (DIB) and Defense Science Board (DSB) findings will be addressed.
EVALUATION PROCEDURE:

Grades will be based on student homework, class contributions, student project, student project presentation, submission of articles and the final exam in the following proportions:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Contribution (Contributions In Addition to the Six Articles*)</td>
<td>10 %</td>
</tr>
<tr>
<td>Homework</td>
<td>10 %</td>
</tr>
<tr>
<td>Six Articles*</td>
<td>10%</td>
</tr>
<tr>
<td>Project</td>
<td>20 %</td>
</tr>
<tr>
<td>Student Project Presentations</td>
<td>10 %</td>
</tr>
<tr>
<td>Final Take Home Exam</td>
<td>40 %</td>
</tr>
</tbody>
</table>

Note: Final exam turn-in date: 3 May 2021

* Articles are submitted on-line. Students can submit their articles at any time. **Note:** All articles will be accompanied with a one-page analysis of each article. Three articles are to be from refereed sources and three can be from any source.
<table>
<thead>
<tr>
<th>Session</th>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25-Jan</td>
<td>Introduction to Project Management</td>
</tr>
<tr>
<td>2</td>
<td>1-Feb</td>
<td>Process Models for Software Development</td>
</tr>
<tr>
<td>3</td>
<td>8-Feb</td>
<td>Establishing Project Foundations</td>
</tr>
<tr>
<td>4</td>
<td>15-Feb</td>
<td>Plans and Planning</td>
</tr>
<tr>
<td>5</td>
<td>22-Feb</td>
<td>Project Planning Techniques</td>
</tr>
<tr>
<td>6</td>
<td>1-Mar</td>
<td>Estimating Techniques</td>
</tr>
<tr>
<td>7</td>
<td>8-Mar</td>
<td>Measuring and Controlling Work Products</td>
</tr>
<tr>
<td>Session</td>
<td>Date</td>
<td>Topic</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>8</td>
<td>15-Mar</td>
<td>Measuring and Controlling Work Processes</td>
</tr>
<tr>
<td>9</td>
<td>22-Mar</td>
<td>Managing Project Risk</td>
</tr>
<tr>
<td>10</td>
<td>29-Mar</td>
<td>Teams, Teamwork, Motivation, Leadership and Communication</td>
</tr>
<tr>
<td>11</td>
<td>5-Apr</td>
<td>Organizational Issues</td>
</tr>
<tr>
<td>12</td>
<td>12-Apr</td>
<td>Future of Software Engineering and Its Impact on Society</td>
</tr>
<tr>
<td>13</td>
<td>19-Apr</td>
<td>Student Presentations</td>
</tr>
<tr>
<td>14</td>
<td>26-Apr</td>
<td>Student Presentations</td>
</tr>
<tr>
<td>15</td>
<td>3-May</td>
<td>FINAL EXAM RETURNED</td>
</tr>
</tbody>
</table>
Blackboard Learn

- Blackboard Learn (previously the *Blackboard* Learning Management System) is a virtual learning environment and course management system developed by *Blackboard* Inc.
- Used by George Mason University in support of online learning
- SWE 625 Course information and assignments are contained on Blackboard Learn
- SWE 625 online synchronous courses will use Blackboard Collaborate Ultra
- Key files are “Weekly Lectures” and Tools (Blackboard Collaborate Ultra)
Blackboard Learn

• Conduct of the Course SWE 625
  – SWE 625 is being offered as an online synchronous course this semester, Spring 2021
  – The normal class time are on Monday, 7:20 – 10:00
  – The course lectures will be conducted virtually using Blackboard Collaborate Ultra which can be accessed via logging into the 625-class using Blackboard and going to Tools at the start of class.
  – Students can gain access to the lecture and class materials (e.g., assignments) prior to the class via logging into the 625-class using Blackboard and going to Lectures
Lesson 1 Assignment

1. Study Chapter 1 in Textbook 1

2. Study Chapter 1 in Textbook 2

3. Read & Provided Comments (1 Pg.) A Retrospective View of the Laws of Software Engineering, Capers Jones, 2017

4. Answer questions: 1.1, 1.3, 1.17 (Textbook No 1)

5. Answer question: 1.4(a) (Textbook No. 2)

All assignments are to be turned in by the time of the next class period using the Blackboard Learn.

All articles should be accompanied with approximately a one-page analysis (i.e. 50% on the content and 50% on your view of the article).
Blackboard Learn (Lecture 1 Asset Library)

Spring
Blackboard Learn

Fall
Launch File
Sign-up for SWE 625 and Come Expecting to Experience One of the Most Career Beneficial Classes

• Successfully managing software intensive projects is a priority for the industrial, government and academic organizations

• The ubiquity of software and its critical role require fundamental shifts in software engineering management and engineering to maintain competitive advantage

• The course helps participants to rapidly deploy innovation with confidence within this shifting landscape by:
  • Applying new principles in software engineering management for software intensive systems
  • Developing new practices for enabling business/mission capability with software innovation

• Equips students in applying new management techniques in today’s competitive job market