Updated 11-13-08

ARIZONA STATE UNIVERSITY
PROPOSAL TO ESTABLISH A NEW UNDERGRADUATE DEGREE

This template is to be used only by programs that have received specific written approval from the Provost's office to proceed with internal proposal development and review. A separate proposal must be submitted for each individual new degree program.

DEGREE PROGRAM INFORMATION

College/School(s) offering this degree: College of Technology and Innovation

Unit(s) within college/school responsible for program: Department of Engineering

Proposed Degree Name: Software Engineering

Undergraduate Degree Type: BS-Bachelor of Science

Proposed title of major: Software Engineering

Is a program fee required? Yes ☒ **No** □

Requested effective term: Fall and year: 2010
(The first semester and year for which students may begin applying to the program.)

PROPOSAL CONTACT INFORMATION

(Person to contact regarding this proposal)

Name: Chell Roberts **Title:** Chair

Phone: 480 727 1353 **email:** Chell.Roberts@asu.edu

DEAN APPROVAL

This proposal has been approved by all necessary unit and College/School levels of review, and the College/School(s) has the resources to offer this degree program. I recommend implementation of the proposed degree program. (*Note: An electronic signature, an email from the dean or dean's designee, or a PDF of the signed signature page is acceptable.*)

College Dean name: Keith Hjelmstad

College Dean signature ______________________________ **Date:** __________

College Dean name:
(if more than one college involved)

College Dean signature ______________________________ **Date:** __________

CIP Code: 14.0903 Computer Software Engineering
(To be determined by the Office of the Executive Vice President and the Provost of the University)

ARIZONA STATE UNIVERSITY
PROPOSAL TO ESTABLISH A NEW UNDERGRADUATE DEGREE

This proposal template should be completed in full and submitted to the University Provost's Academic Council [mailto:curriculum@asu.edu]. It must undergo all internal university review and approval steps including those at the unit, college, and university levels. A program **may not** be implemented until the Provost's Office notifies the academic unit that the program may be offered.

DEGREE PROGRAM INFORMATION

Undergraduate: BS-Bachelor of Science

Proposed title of major: Software Engineering

1. PURPOSE AND NATURE OF PROGRAM

A. **Brief program description** (This is a catalog type description. Include the distinctive features of the program that make it unique. Do not include program or admission requirements.)

The program blends engineering, computing, project leadership, and technology education. Software systems are complex, often including in excess of a million lines of code. Graduates of the BS Software Engineering will possess knowledge and skills of a defined engineering approach to complex software systems analysis, planning, design, and construction. The program is a unique project-driven curriculum, establishing a new model for Software Engineering education. The program is built around the concepts of engaged-learning, discovery-based education and learn-by-doing. Students do projects in every semester of the program to provide emphasis in communication, team-work, innovation, critical thinking, and professionalism. Students have a high degree of flexibility in designing their course of study; they select a software engineering application area as their primary focus, and they may obtain inter-disciplinary knowledge through a secondary related area of their choosing. Students learn computing and software engineering fundamentals in the first two years of the curriculum through projects with tightly controlled learning outcomes. Junior and senior year projects become more typical of a professsional computing environment as they participate in on-going projects whose outcomes are fused with learning software engineering principles and content in their primary focus area. The curricular structure in the upper-division is uniquely defined to accommodate learning software engineering concepts and focus area concepts together, and to provide a capstone project experience.

2. STUDENT LEARNING OUTCOMES AND ASSESMENT

A. **List the knowledge, competencies, and skills** students should have when they graduate from the proposed degree program. (You can find examples of program Learning Outcomes at (http://www.asu.edu/oue/assessment.html)

The program is designed to be accredited by the Accreditation Board for Engineering and Technology (ABET) BS Software Engineering criteria. Educational Learning Objectives and Outcomes are defined based on ABET definitions.

Program objectives describe the career and professional accomplishments that the program is preparing graduates to achieve.

The BS Software Engineering Learning Objectives: While a student at Morehouse College, Dr. Martin Luther King Jr. wrote in a campus publication: "We must remember that intelligence is not enough. Intelligence plus character - that is the goal of true education." The BS Software Engineering embraces this broader view of education. The program is designed to be flexible so that students can individualize to their own specific objectives. The course of study is project-driven where at the lower-division students complete tangible projects with tightly controlled learning outcomes, and at the upper-division projects more closely match the profession. The program's goal is to provide a

professionally guided education in software engineering that prepares graduates to assume careers in which they:

1. Solve computing problems from algorithmic and systems perspectives, understanding the nature of computing, and the broad reach of computing's impact on society; how to ethically apply concepts, tools and technologies they've learned, and how to synthesize innovative computing processes and solutions.

2. Design, analyze, implement, and validate complex software systems utilizing layers of hardware and software; individually and as effective team members, and in a manner reflecting appropriately adapted computing practice and problem solving techniques.

3. Advance their career by valuing and continually upgrading professional, communication, analytic, and technical skills, as well as striving to acquire a deeper understanding of the business, global, environmental, ethical and societal impacts of their career and the technologies they apply and develop.

Program outcomes identify what students know and are able to do at the time of graduation. Outcomes for the program reflect the developmental nature of student learning and as such include levels indicating deeper knowledge in the outcome area. This approach is based on levels in the learning model developed at Alverno College and adapted to the BS Engineering program that is also offered by the Department of Engineering at ASU Polytechnic.

Program outcomes for the BS Software Engineering:

1. Design. An ability to design a digital or software system, component, or process to meet desired needs within realistic constraints.
Level1. Recites the steps and information flow in the engineering, digital system and software design process and uses organizational or technical tools in each step.
Level 2. Given a problem definition, uses a design process and tools to produce a documented solution design, including a prototype; and explains how the design meets the constraints and criteria.
Level 3. Evaluates design process and resulting design quality relative to appropriate criteria, and suggests improvements.
Level 4. Customizes design process, related artifacts, and tools for specific problem contexts.

2. Computing Practice. An ability to employ knowledge, methods, skills, modern software engineering tools, and team-based approaches appropriate for interdisciplinary and multi-cultural settings in arriving at computing problem solutions.
Level 1. Describes the essential elements of team-based computing practice.
Level 2. Given a problem statement, works within a team to develop a plan and identify the tools and methods that will produce a technical solution.
Level 3. Evaluates the effectiveness of the planning process, teamwork and tools used to produce a solution to a computing problem.
Level 4. Adapts the planning process, team work approach, tool sets, and methods to a problem context, and then professionally applies them to obtain optimal (defensible) solutions to computing problems.

3. Critical Thinking and Decision Making. An ability to think critically, clearly identifying and using evidence, criteria, and values in the decision making process.
Level 1. Articulates the critical thinking process.
Level 2. Identifies assumptions, criteria, and evidence to make informed decisions.
Level 3. Evaluates alternative perspectives, contexts, and the quality of evidence in making informed judgments.
Level 4. Examines and cultivates a value system to make informed decisions.

4. Professionalism. An understanding of professional and ethical responsibility, a commitment to on-going professional competence and possession of basic professional and organizational success skills.
Level 1. Identifies professionally appropriate behavior. Appreciates engineering and computing as a learned profession, and possesses daily success skills.
Level 2. Accepts responsibility for personal education. Understands the major professional and ethical responsibilities of computing professionals. Understands the major computing specialties, corporate structures and purposes.
Level 3. Uses accepted ethics and concepts to guide decisions, and formulate a career path that accounts for current trends in technology and society.
Level 4. Effectively guide activities to maintain professional competence and reputation.

5. Perspective. An understanding of the role and impact of engineering and computing technology in business, global, economic, environmental, and societal contexts.
Level 1. Understands that technological change and development have both positive and negative effects.
Level 2. Identifies and evaluates the assumptions made by others in their description of the role and impact of engineering and computing on the world.
Level 3. Selects from different scenarios for the future and appropriately adapts them to match current technical, social, economic and political concerns.
Level 4. Has formed a constructive model for the future of our society, and makes life and career decisions that are influenced by the model.
6. Problem Solving. An ability to identify, formulate, and solve problems whose solutions require digital and software systems using the steps of an analytical problem solving approach.
Level 1. Articulates the problem solving process by making explicit the steps taken to realize a solution.
Level 2. Performs all steps of the problem solving process including conceptualization, elaboration, realization and analysis in both closed and open-ended design and analysis problems.
Level 3. Analyzes, selects, uses, and evaluates various methods and frameworks for developing solutions to computing problems.
Level 4. Adapts tools, methods and frameworks of problem solving to a variety of computing problems requiring a collaborative teamwork approach.

7. Communication. An ability to communicate effectively.
Level 1. Recognizes and describes individual processes used in various modes of communication.
Level 2. Uses a process to develop appropriately structured communications.
Level 3. Purposefully applies communication strategies to interact meaningfully with the audience.
Level 4. Selects and adapts communication strategies to fully engage the audience.

8. Technical Competence. An ability to apply, evaluate and adapt knowledge of mathematics, software, computer science, computer systems, and software engineering in generating complex solutions to problems in a software engineering application area.
Level 1. Verbally and mathematically communicates the fundamental software, computer science, computer systems and software engineering principles, methods, and tools underlying computing solutions.
Level 2. Applies fundamental computing and software engineering principles, methods, and tools to construct solutions to computing problems using structured approach including analysis, modeling, prototyping, and validation.
Level 3. Evaluates, selects and applies appropriate fundamental computing and software engineering principles, methods, and tools relevant to characteristics of a software engineering application area in a structured manner.
Level 4. Selects, adapts, and applies appropriate fundamental computing and software engineering principles, methods and tools to solve complex multidisciplinary computing problems.

Technical Competence (Outcome 8) is the one directly assessing technical proficiency in the software domain. For this reason the outcome is further elaborated through detailed outcomes in computing fundamentals and software engineering:

8.1 Software and Computer Science Fundamentals. Apply knowledge of a programming language, related tools, and execution environment to define problems and create effective software solutions, using appropriate language mechanisms, data structures, algorithms, and software development processes.

Level 1. Communicates the tools and steps involved in creating and executing a program in a programming language, and communicates the programming language building blocks for program construction.

Level 2. For a stated programming problem, applies knowledge of programming, software development processes, and algorithm analysis to create a software solution that utilizes language facilities and abstraction in creating effective algorithms, program structures, information structures and control structures.

Level 3. Analyzes, evaluates, and applies alternative program structures and algorithms in creating software to solve a problem in a high-level language; utilizes language execution environment; validates solutions and processes to determine whether goals and constraints are met.

Level 4. Evaluates alternative language, development and execution environments to match the needs of a software solution; tailors the steps of program design, development and deployment to solve problems within solution constraints. Selects, modifies and applies alternative software processes to match specific domain or contextual needs of a problem.

8.2 Computer Systems Fundamentals. An ability to apply knowledge of computer organization and architecture to match a computer systems problem solution to device characteristics. The ability to develop a software solution that utilizes specific components within their effective performance capabilities.

Level 1. Communicates how processors, memories and peripheral devices are interconnected and work together to function as a computer.

Level 2. Uses knowledge of organization and instruction-set to consider trade-offs between size and speed of major components in realizing solutions that make effective use of a computer.

Level 3. Analyzes and applies system component performance information to realize solutions on systems that cannot be considered general purpose computers.

Level 4. Customizes devices and their components based on functional, performance, size and power requirements to arrive at a system configuration that matches the needs of the application and underlying software.

Technical Outcomes in Software Engineering

8.3. Software Design. Understands the importance of modern software architecture principles, applies design techniques to create models to represent software architecture, and analyzes tradeoffs of design decisions on system and operational constraints. Understands broader impacts of the design process, including economic and regulatory considerations. Able to utilize modeling languages and customized notations based on the domain of application.

Level 1. Understands the need for, and perspectives of, software system models. Able to read and understand software models created in a general purpose modeling language.

Level 2. Able to create models in at least two general-purpose modeling languages and one domain-specific modeling extension.

Level 3. Applies different modeling languages and processes based on attributes of the system requirements. Able to trace and validate design artifacts with respect to requirements.

Level 4. Analyzes, evaluates and adapts a software design methodology based on system and operational constraints. Synthesizes the use of different modeling perspectives to communicate the technical, economic, regulatory, etc. impacts of the architecture.

8.4 Software Process. Understands and applies software lifecycle process models in the context of a business process model. Understand and applies modern techniques within each lifecycle phase (requirements, analysis and design, implementation, testing, maintenance, and evolution). Verifies and validates the extent to which solutions meet intended goals, and is able to use the results of evaluations to improve the solution. Applies an instrumented software development process where measures and evaluations are aimed at process improvement for both individual and team-based development. Understands basic concepts in software project management and the relationship of management to implementation.

Level 1. Articulates a software lifecycle process including lifecycle activities and related project management structures, artifacts, quality measures, and configuration needs.

Level 2. Applies a specified program construction and software lifecycle process to elicit requirements, define the problem, analyze, design and realize an appropriate solution; and produce appropriate software artifacts that meet customer constraints. Applies a structured individual software process.

Level 3. Applies an instrumented software process collecting information on lifecycle activities and the system being developed. Evaluates measures in the context of a development and uses the results to determine attainment of solution goals. Effectively applies a team-based software process.

Level 4. Analyzes, evaluates and adapts a software process utilizing process measures and software quality measures to improve the team-based software process and product.

B. **Describe the plan and methods to assess** whether students have achieved the knowledge, competencies and skills identified in the Learning Outcomes. (You can find examples of assessment methods at (http://www.asu.edu/oue/assessment.html)

The assessment plan has the dual purpose of determining student achievement of educational objectives and outcomes, as well as evolution and improvement of the program. Program assessment is accomplished by periodically collecting information from a variety of sources, which is evaluated by program administration and faculty to determine appropriate curriculum changes. Assessment information includes: student and faculty course assessment, graduating student survey, graduating student interview, student generated portfolios of outcome achievement, alumni survey (three and five year out), alumni employer survey, and industry council input. The professional organization Institute for Electrical and Electronics Engineering (IEEE) Computer Society supports certification exams in software engineering that may be used as further input. Information from these sources will be evaluated to determine student achievement of program outcomes and learning objectives, and it will be evaluated in conjunction with information received from the program's industry advisory council to refine and evolve program curriculum, learning objectives and outcomes. Program faculty and administration are responsible for conducting regular evaluations, as well as initiating and documenting appropriate curricular action.

Student assessment is accomplished by a combination of student portfolio evaluation, project spine evaluation, and traditional classroom assessment methods include the grading of exams, quizzes, (in-class and take-home) laboratory assignments, etc. Course level outcomes will be tied to program level outcomes. For student portfolios, the software engineering faculty will mimic the practices defined and implemented by the recently accredited General Engineering program in the same unit. Students submit portfolios for each of the outcomes defined above in 2A; they are required to self-assess through the accumulation and presentation of evidence of their achievement of target levels for each outcome. Students are required to achieve at least level 2 in all outcomes, and level 3 in at least one of the eight outcomes. Project-related activities are assessed by multiple faculty involved in the projects at each year. Project activities have faculty coordinators and faculty mentors; all faculty involved in working with a project team assess the students on the team. Project team activities also allow for peer review, particularly in the upper division. Finally, the senior year project experience is designated as the capstone project experience where students work on externally sponsored projects and are assessed by the faculty, student peers, and the external sponsors.

3. CURRICULUM OF THE PROPOSED PROGRAM

Total credit hours must be 120 to include: first year composition, general studies, core/required courses, program specific electives, and any additional requirements.

A. **Major Map.** Please prepare and attach a Major Map. If there are concentrations in this degree program, prepare a separate Major Map for each one. (Examples of Major Maps can be found at http://provost.asu.edu/curriculum)

See attached.

B. **Total credit hours required for this program:** 120 semester hours.

C. **Core/Required Courses.**

i. Total required and/or core course credit hours: 56 semester credit hours.

ii. List the name, prefix, and credit hours for each required/core class for this program
Introduction to Engineering and Computing Design I, EGR 101, 3 credits
Introduction to Engineering and Computing Design II, EGR 102, 3 credits
Computer Systems Fundamentals I, SER 232, 3 credits
Software Engineering I, SER200, 1 credit (taught as engineering module, see EGR program)
Software Enterprise I, SER 215, 3 credits (must be taken with SER 200, SER221)
Software Enterprise II, SER 216, 3 credits (must be taken with SER 222)
Software Fundamentals I, SER 221, 3 credits
Software Fundamentals II, SER 222, 3 credits
Discrete Structures and Applied Computing Theory, SER 203, 3 credits
Operating Systems and Networking, SER 234, 3 credits
Software Enterprise III, SER 315, 2 credits (must be taken with SER 325)
Software Enterprise and Focus Project I, SER 325, 2 credits
Software Enterprise IV, SER 316, 2 credits (must be taken with SER 326)
Software Enterprise and Focus Project II, SER 326, 2 credits
Software Enterprise V, SER 415, 2 credits (must be taken with SER 425)
Software Enterprise and Focus Project III, SER 425, 2 credits
Software Enterprise VI, SER 416, 2 credits (must be taken with SER 426)
Software Enterprise and Focus Project IV, SER 426, 2 credits

At the upper-division, students select a software engineering application area as their primary focus. Each focus area consists of four courses, each course having three semester credits. Students start with Software Systems SER 321, 3 credits; then continue through three more primary focus area courses. Students enrolled in a focus area course (3 credits) must be concurrently enrolled in the corresponding software enterprise course (2 credits) and in the software enterprise & focus project course (2 credits). Thus, each semester during the junior and senior year students are enrolled in 7 semester credits totaled from primary focus, project, and software enterprise. The projects fuse content from the software enterprise with the focus area, and are coordinated so that projects and project teams co-mix juniors and seniors. In the six-semester software enterprise sequence, students are exposed respectively to:

Software tools, processes, teams and personal software process
Software reliability and testing
Software requirements elicitation,
Project construction/realization,
Software project design,
Software quality assurance and testing,
Software maintenance and evolution,
Software process and project management.

Software Engineering accreditation requires students to develop technical expertise in at least one application area. The degree program mechanism we will use to address this

requirement is the primary focus area, as described above. The first primary focus area leverages course material and faculty expertise in the area of
Distributed and Web-Based Systems. Future focus areas may be introduced based on faculty expertise, current trends, and availability of resources.

D. **Program Specific Electives.**
 i. Total required program elective credit hours: 12
 ii. List the name, prefix, and credit hours for any program specific electives for this program:
 There are 12 semester hours of upper-division secondary focus. Through consultation with a program faculty member, students design the secondary focus to meet their unique educational objectives while adhering to the constraint that the secondary focus consist of related upper-division courses in an area that is relevant to Software Engineering and the student's educational goals. Examples may include a focus area within engineering, such as robotics, or electrical. The secondary area may also be in management, business, finance, gaming, or art/animation. This flexibility opens career pathways requiring multi-disciplinary or novel combinations of expertise.

E. **Additional Program Requirements, if any.** List and describe any capstone experiences, milestone, and/or additional requirements for this degree program:

The program's capstone experience is built into the Software Enterprise project sequence, which begins in the third semester. In the software enterprise, students participate on teams to (re)define and complete a project. Students are exposed to multi-disciplinary project teams and participate on teams in which they are assigned various roles (tester, team-leader, project manager, designer, requirements elicitor, and programmer, for example).

F. **Are any concentrations to be established under this degree program?** □ Yes ☒ No

4. NEW COURSE DEVELOPMENT

A. Will a new course prefix(es) be required for this degree program? Yes ☒ No □
If yes, complete the request for establishment of a new prefix for each prefix and submit with this proposal.

B. New Courses Required for Proposed Degree Program. List all new courses required for this program, including course prefix, number and course description.

All of the courses listed in section 3.C are new courses having the course number and title as listed in that section.

SER 200 Software Engineering I (module)
Overview of software lifecycle process models and process phases; design and implementation of software from an engineered perspective; stakeholders of the software process. (1 credit, no preq)

SER 232 Computer Systems Fundamentals I
Logic design and computer organization; number systems and arithmetic, boolean algebra; digital system compo- nents; assembly language and instruction set concepts and application. (3 credits; preq EGR 101)

SER 221 Software Fundamentals I
Object-oriented design, modeling, and programming; inheritance and polymorphism; specification, application and implementation of data structures: strings, stacks, queues, trees, hash tables. Introduction to time and space complexity, recursion, searching, and sorting. (3 credits; preq EGR 102, coreq SER 215)

SER 222 Software Fundamentals II
Programming language constructs (generics, parameter passing, lambda expressions, closures, etc.); programming paradigms (procedural vs. declarative, compiled vs. interpreted, etc.); runtime environment management; exposure to multiple languages. (3 credits; preq SER 221, coreq SER216).

SER 234 Operating Systems and Networking
Fundamentals of operating systems, process management, scheduling and synchronization techniques, memory and file management, protection and security issues. Computer networking for application, transmission control and network layers using the Internet protocols as a model; reliability and security. (3 credits; preq SER 221)

SER 215 Software Enterprise I (Project)
Software engineering; personal software processes for individual professionalism; time and defect estimation, yield, and productivity. Software tools. Project-based. (3 credits, preq EGR102, coreq SER 221, pre/coreq SER 200)

SER 216 Software Enterprise II (Project)
Software engineering; software processes for teamwork and communication; software development tools; software verification and validation processes and technologies. Project-based. (3 credits, preq SER200, SER215, coreq SER 222).

SER 203 Discrete Structures and Applied Computing Theory
Algorithm design and analysis, classes of computability, searching, sorting graph specification and analysis; Language specification and recognizers, regular and context-free languages; introduction to decidability and function theory; computing representations of logical reasoning, functions, number systems, sets, relations, and induction; introduction to combinatorics and finite probability. (3 credits; preq MAT 265, EGR102).

SER 321 Software Systems I
Software architecture and design; architectural styles and design patterns; design process as part of the software lifecycle process; logical and physical models of design; components, connectors, and assemblies; distributed design and implementation constructs including sockets, messages, remote method invocation, and other higher-level application protocols; concurrency, multithreading, and multi-process systems. (3 credits, preq SER216, SER222, SER232).

Junior/Senior Year Project Spine

SER 325 Focus Project I
Project-based. Work with existing codebase to create a scalable software system or application in the primary focus area. Configuration Management. Specifications for new features. (2 credits; coreq SER315).

SER 315 Software Enterprise III
Software engineering; understanding, maintaining, and evolving existing software; requirements specifications. (2 credits; coreq SER305, and SER 321 or SER331)

SER 326 Focus Project II
Project-based. Continue project from SER305. Software builds, packaging, and release management. Traceability, defect management, software maintenance. (2 credits; preq SER325, coreq SER316).

SER 316 Software Enterprise IV
Software engineering; software development best practices, unit testing, refactoring, defensive programming, measures and metrics, static and dynamic analysis. (2 credits; preq SER315, coreq SER326 and primary focus area course)

SER 425 Focus Project III

Project-based. Work with external customer to elicit business need and software features. Prototype solution, deliver software requirements specification (SRS). (2 credits; coreq SER415 and primary focus area course).

SER 415 Software Enterprise V (L)

Software engineering; requirements elicitation and capture; business domain analysis; software project management fundamentals (2 credits; coreq SER425)

SER 426 Focus Project IV

Project-based. Continuation of SER405. Quality management planning and test scripting; verification and validation; governance (2 credits; coreq SER416 and primary focus area course).

SER 416 Software Enterprise VI

Software engineering; quality assurance; business models for delivered software; software deployment and transition to operation. (2 credits; coreq SER426)

JUNIOR/SENIOR YEAR PRIMARY FOCUS AREA COURSES

The initial primary focus area will be in Distributed and Web-based Systems; this is an area of deep faculty expertise in the department. Additional focus areas are expected with program growth. The Distributed and Web-based Systems area will be comprised of SER321 (above) plus the four new SER courses below plus the existing CST433 Database Technology course. The student will take SER 321 plus three of the other five courses to complete the four-course focus area.

SER 471 Web Application Development

Fundamentals of web application development. Programming language web development frameworks, and runtime environment paradigms. Software engineering processes and best practices for web applications in healthcare, education, finance, etc. Principles of usability, accessibility, security, privacy, and user responsiveness. (preq SER 321)

SER 472 Web Systems

Architecture and design principles of the scalable web. Reliability, availability, fault tolerance, and security from a software systems perspective. Transitioning software to enterprise operation. Software design for performance and scalability including horizontal replication, vertical fragmentation, and caching infrastructures. An introduction to modern scalability technologies such as cloud computing, the semantic web, and service-oriented computing. (preq SER 321)

SER 473 Middleware for Enterprise Architectures

Principles and technologies in software middleware. Location transparency, identity management, and enterprise component models. Messaging-oriented middleware (MOM), Services-Oriented Architecture (SOA), REpresentational State Transfer (REST), and Peer-to-Peer (P2P) architectural styles. Issues in adopting modern middleware architectures in the enterprise. (preq SER 321)

SER 477 Mobile Computing and Applications

Mobile devices and network-centric applications for them. Analysis, design and creation of mobile applications using separate development and target environments; simulators, emulators and devices. Frameworks for device services such as, telephone, camera, audio, video, location, network, personal information, and device orientation. User-interfaces for screen-limited and touch devices; synchronizing devices and their information stores; operating systems and language runtime environments for limited devices. (preq SER 321)

5. **PROGRAM NEED.** Explain why the university needs to offer this program (include target audience and market).

   The BS Software Engineering will increase ASU's ability to meet the regional need for computing professionals and facilitate a stronger regional computing industry. The program's unique project-driven, professional, and interdisciplinary approach will attract a new set of students than are currently attracted to ASU computing programs, and it will produce graduates with a knowledge and skill-set that differs from existing ASU computer science and information systems programs. The program will add quality and capacity to ASU's computing and engineering offerings through a differentiated curricular approach and outcome-set.

   The Bureau of Labor Statistics Occupational Handbook (http://www.bls.gov/oco/home.htm) projects software engineers as among the fastest growing occupations in the period 2008-2018. The program prepares graduates for three of the top eight fastest growing occupations requiring a bachelor's degree (software engineering applications, software engineering systems, and network / data systems analysis). In these areas, occupational growth (nationally) is forecast to reach 1.6 million from the current level of 1.2 million. Starting salaries for baccalaureate graduates are among the highest nationally (http://money.cnn.com/2005/02/09/pf/college/nace_survey/index.htm) (http://www.payscale.com/research/US/Job=Software_Engineer_%2F_Developer_%2F_Programmer/Salary) .

6. **IMPACT ON OTHER PROGRAMS.** List other academic units that might be impacted by the proposed program and describe the potential impact (e.g., how the implementation of this program might affect student headcount/enrollment, student recruitment, faculty participation, course content, etc. in other programs). Attach letters of collaboration/support from impacted programs.
   The Ira Fulton School of Engineering Tempe campus offers the BS Computer Science with concentration in Software Engineering. The unique mode of delivery, the engineering nature, and the interdisciplinary nature of the BS Software Engineering program as proposed to be offered on the Polytechnic campus will add quality, variety and capacity to ASU offerings in the computing area. There will be little or no impact on related Tempe campus programs.

7. **PROJECTED ENROLLMENT** How many new students do you anticipate enrolling in this program each year for the next five years? Please utilize the following tabular format.

   Program enrollment projections are based on

5-YEAR PROJECTED ANNUAL ENROLLMENT					
	1st Year	**2nd Year** (Yr 1 continuing + new entering)	**3rd Year** (Yr 1 & 2 continuing + new entering)	**4th Year** (Yrs 1, 2, 3 continuing + new entering)	**5th Year** (Yrs 1, 2, 3, 4 continuing + new entering)
Number of Students Majoring (Headcount)	30	65	110	160	215

8. **ACCREDITATION OR LICENSING REQUIREMENTS (if applicable).** Provide the names of the external agencies for accreditation, professional licensing, etc. that guide your curriculum for this program, if any. Describe any requirements for accreditation or licensing.
   The Accreditation Board for Engineering and Technology (ABET) *Criteria for Accrediting Engineering Programs* (see: http://www.abet.org) include general and program criteria for Software Engineering. The program has been designed to satisfy these criteria, and accreditation will be sought the year subsequent to the first graduating class (year four). The ABET criteria includes sections on: Students (advising, student performance, outcome achievement, and program audit), Educational Objectives, Program Outcomes, Continuous

Improvement, Curriculum, Faculty, Facilities, Support, and Software Engineering Program Specific Criteria.

9. FACULTY and STAFF

a. **Current Faculty**. List the name, rank, highest degree, area of specialization/expertise and estimate of the level of involvement of all current faculty who will teach in the program.

Ashish Amresh, Assistant Professor, Ph.D. (ABD, expect September 2010), Gaming, Computer Science,
Ramon Anguamea Lara, Lecturer, Master of Computing Studies, Embedded Systems,
Srividya Kona Bansal, Ph.D. Computer Science, Software Engineering,
John Femiani, Assistant Professor, Ph.D. Computer Science, Visual Analytics,
Kevin Gary, Associate Professor, Ph.D. Computer Science, Software Engineering,
Arbi Ghazarian, Assistant Professor, Ph.D. Computer Science, Software Engineering
Ben Huey, Associate Professor, Ph.D. Electrical Engineering, Embedded Systems,
Timothy Lindquist, Professor, Ph.D. Computer Science, Software Engineering,
Bruce Millard, Associate Professor of Practice, Ph.D. Computer Science, Operating Systems and Networking
Anshuman Razdan, Associate Professor, Ph.D. Computer Science, Visual Analytics
Richard Whitehouse, Lecturer Sr., MS Computer Science, Software Engineering/Mobile Computing

The BS in Software Engineering is envisioned as the growth program in computing within the Department of Engineering and College of Technology and Innovation. All listed faculty will participate extensively to this growth vision, while at the same time supporting the existing BS in Computer Systems. Recent hires (Ghazarian, Bansal, and Amresh) are particularly suited to supporting the new program.

b. **New Faculty.** Describe the new faculty hiring needed during the next three years to sustain the program. List the anticipated hiring schedule and financial sources for supporting the addition of these faculty.

No new faculty are required.

c. **Administration of the program.** Explain how the program will be administered for the purposes of admissions, advising, course offerings, etc. Discuss the available staff support.

The program will be administered by the Department of Engineering at the Polytechnic campus. The Chair, Dr. Chell Roberts and departmental support staff will provide administrative oversight. Advising will be provided by the College of Technology and Innovation using the same model currently in use for all other programs in the college. Admission, registration, course scheduling, and graduation (audit) support will be provided as is currently provided for the other programs in the Department of Engineering – through a combination of support at the departmental, college and university levels.

10. RESOURCES (necessary to launch and sustain the program)

a. Describe any new resources required for this program's success, such as new support staff, new facilities, new library resources, new technology resources, etc.

Existing resources will be reassigned from other functions within the College of Innovation and Technology (and Department of Engineering) to provide support to sustain the BS Software Engineering program.

b. Explain where you will get the resources to support this program.

APPENDIX
OPERATIONAL INFORMATION FOR UNDERGRADUATE PROGRAMS

(This information is used to populate the Degree Search /catalog website.)

1. Contact and Support Information

Office Location (Building & Room): Santan 230

Campus Telephone Number: 480 727 2727

Program email address: egr@asu.edu

Program website address: http://technology.asu.edu/engineering

2. Additional Program Description Information

A. Additional program fee required for this program? Yes ☒ No □

B. Does this program have a second language requirement? Yes □ No ☒

3. Career Opportunities & Concentrations

Provide a brief description of career opportunities available for this degree program. If program will have concentrations, provide a brief description for each concentration.

Software engineers solve a broad set of interdisciplinary problems and apply new technologies to improve the quality of life. Graduates design and engineer innovative systems that may include mechanical, and electronic components that interact with software. The BS Software Engineering is a unique program where students learn by solving engaging projects, commonly as a member of a development team. The program prepares graduates for advanced study in computing or an allied field, or to enter the computing profession, most commonly as an application software engineer. According to the Bureau of Labor Statistics, software engineers are highly paid and there is significant growth in the number of employment opportunities. Some software engineering jobs may include creating video games, designing embedded systems, developing distributed applications, creating Web applications, or working on data, network, security, or computer system administration.

4. Additional Admission Requirements

If applicable list any admission requirements (freshman and/or transfer) that are higher than and/or in addition to the university minimum undergraduate admission requirements.)

none

5. Keywords

List all keywords used to search for this program. Keywords should be specific to the proposed program.

Software, Engineering, Computing, Technology, Gaming, Embedded Systems, Web, Mobile Computing, Graphics Programming, Animation, Simulation

6. Area(s) of Interest

A. Select one (1) primary Area of Interest from the list below that applies to this program.

- □ Architecture, Construction & Design
- □ Artistic Expression & Performance
- □ Biological Sciences, Health & Wellness
- □ Business, Management & Economics
- □ Communication & Media
- ☒ Computing & Mathematics
- □ Education & Teaching
- □ Engineering & Technology

□ Environmental Issues & Physical Sci
□ Interdisciplinary Studies
□ Languages & Cultures
□ Law & Justice
□ Social Science, Policies & Issues

B. Select any additional Areas of Interest that apply to this program from the list below.
□ Architecture, Construction & Design
□ Artistic Expression & Performance
□ Biological Sciences, Health & Wellness
□ Business, Management & Economics
□ Communication & Media
□ Computing & Mathematics
□ Education & Teaching
□ Environmental Issues & Physical Sci
☒ Engineering & Technology
☒ Interdisciplinary Studies
□ Languages & Cultures
□ Law & Justice
□ Social Science, Policies & Issues

ARIZONA STATE UNIVERSITY

Major Map: Software Engineering
Bachelor of Science (B.S.)

College of Technology and Innovation, Polytechnic Campus
Catalog Year: 2011-2012

			Competed Transfer Pathway: ☐ MAPP ☐TAG ☐ATP ☐None		Completed General Education: ☐AGEC ☐IGETC/CSUGE ☐None
Course Subject and Title *(courses in bold/shading are critical)*	Hrs.	Upper Division	Transfer Course/Grade	Minimum Grade if Required	Additional Critical Requirement Notes
TERM ONE: 0-15 CREDIT HOURS					
ASU 101: The ASU Experience	1	☐			• ASU 101 is for ASU freshman students only Not required of transfer students • An SAT, ACT, Accuplacer, or TOEFL score determines placement into first-year composition courses • ALEKS placement determines mathematics course • EGR 104 may be taken in term 2 if PHY 121/122 taken in term 1. • *Literacy Designation waiting for approval.
MAT 265: Calculus for Engineers I (MA) or pre-requisite	3	☐		Grade of C	
EGR 101: Intro to Engineering & Computing Design I	3	☐		Grade of C	
EGR 104: Critical Inquiry for Engineering *(L)	3	☐		Grade of C	
ENG 101 or 102: First-Year Composition or ENG 105: Advanced First-Year Composition or ENG 107 or 108: English for Foreign Students	3	☐		Grade of C	
TERM TWO: 16-30 CREDIT HOURS					
SER 232 Computer Systems Fundamentals I	3	☐		Grade of C	• PHY 121/122 may be taken in term 1 if student has AP credit for MAT 265.
MAT 266: Calculus for Engineers II	3	☐		Grade of C	
EGR 102: Intro to Engineering & Computing Design II	3	☐		Grade of C	
PHY 121/122: University Physics I/Laboratory I (SQ)	4	☐		Grade of C	
ENG 101 or 102: First-Year Composition or ENG 105: Advanced First-Year Composition or ENG 107 or 108: English for Foreign Students	3	☐		Grade of C	
TERM THREE: 31-45 CREDIT HOURS					
SER 221: Software Fundamentals I	3	☐		Grade of C	• Complete First Year Composition requirement by end of semester 3 • SER 200 may be taken in term 1.
SER 215: Software Enterprise I	3	☐		Grade of C	
SER 200: Software Engineering (Module)	1	☐			
SER 203: Discrete Structures and Applied Computing Theory	3	☐			
Humanities (HU) course with cultural (C) awareness area	3	☐			
Humanities (HU) course with historical (H) awareness area	3	☐			
TERM FOUR: 46-60 CREDIT HOURS					
SER 221: Software Fundamentals II	3	☐		Grade of C	
SER 216: Software Enterprise II	3	☐		Grade of C	
SER 234: Operating Systems and Networking	3	☐			
EGR 280: Engineering Statistics (CS)	3	☐			
Social Behavior (SB)	3	☐			
TERM FIVE: 61-75 CREDIT HOURS					
SER 315: Software Enterprise III	2	☒			
SER 325: Software Enterprise & Focus Project I	2	☒			
MAT 343: Applied Linear Algebra	3	☒			
SER 321: Software Systems	3	☒			
Secondary Focus Area (select with advisor)	3	☐			
HST 318: History of Engineering (SB & G)	3	☒			
TERM SIX: 76-90 CREDIT HOURS					
SER 316: Software Enterprise IV	2	☒			
SER 326: Soft Enterprise & Focus Project II	2	☒			
SER 3[3,4,5,6]2: Primary Focus Area	3	☒			
Secondary Focus Area (select with advisor)	3	☐			
Lab Science II (SQ or SG)	4	☐			
TERM SEVEN: 91-105 CREDIT HOURS					
SER 415: Software Enterprise V *(L)	2	☒			• *Upper Division Literacy Designation approval needed.
SER 425: Soft Enterprise & Focus Project III * (L)	2	☒			
SER 4[3,4,5,6]1: Primary Focus III	3	☒			
Secondary Focus Area (select with advisor)	3	☒			
Lab Science Elective: Choose 1: PHY 131/132: University Physics II/Laboratory II (SQ) BIO 182: General Biology II (SG) CHM 113: General Chemistry I (SQ) GLG 101/103: Intro to Geology I (Physical)/Laboratory (SQ & G)	4	☐			
TERM EIGHT: 106-120 CREDIT HOURS					
SER 416: Software Enterprise VI	2	☒			
SER 426: Software Enterprise & Focus Project IV	2	☒			
SER 4[3,4,5,6]2: Primary Focus IV	3	☒			
Secondary Focus Area (select with advisor)	3	☒			
Math or Science Elective	3	☒			
Humanities (HU) or Social Behavioral Science (SB)	3	☐			

ARIZONA STATE UNIVERSITY

Major Map: Software Engineering
Bachelor of Science (B.S.)

College of Technology and Innovation, Polytechnic Campus
Catalog Year: 2011-2012

Graduation Requirements Summary:

Total Hours (120 minimum)	Total Hrs at ASU (30 minimum)	Hrs Resident Credit for Academic Recognition (56 minimum)	Major GPA (2.000 Min.)	Total UD Hrs (45 minimum)	Total Comm. College Hrs. (64 maximum)

General University Requirements: Legend

- General Studies Core Requirements:
  - Literacy and Critical Inquiry (L)
  - Mathematical Studies (MA)
  - Computer/Statistics/Quantitative applications (CS)
  - Humanities, Fine Arts, and Design (HU)
  - Social and Behavioral Sciences (SB)
  - Natural Science-Quantitative (SQ)
  - Natural Science-General (SG)
- General Studies Awareness Requirements
  - Cultural Diversity in the US (C)
  - Global Awareness (G)
  - Historical Awareness (H)
- First-Year Composition

Additional Notes:

ASU
ARIZONA STATE UNIVERSITY

Course Prefixes/Subjects
Arizona State University

Requested Action

☒ Establish a new prefix (complete section **B only**)

□ Disestablish an existing prefix (complete section **A** only)

□ Change an existing prefix (complete sections **A & B**)

A. Disestablish or Change an Existing Prefix

College/School/Division	Select College/School/Division		
Department/Unit			
Prefix (3 letters)		Prefix name	
Expiration Year		Expiration Term	

B. Establish a New Curriculum or Change an Existing Prefix

From ACD 305-13: Establishing New Course Prefixes (http://www.asu.edu/aad/manuals/acd/acd305-13.html)

Distinctive course prefixes are appropriate to set off the course offerings of one university academic program from others. It is not necessary that the program constitute a formal major or minor, but the prefix should correspond to an element of ASU's curriculum that is administered by a formal programmatic entity, which might be a department, an interdisciplinary committee, or, under unusual circumstances, a college or other unit. Responsibility for administration of courses that are listed under the prefix must be clearly established—that is, responsibility for scheduling classes, assuring the quality of the coursework, ensuring the submission of grade reports, and maintaining student records.

A request for a new prefix should be submitted in a memo of two pages or less. The request should follow the following outline:

- Describe the reason for the request. Why, in particular, is it important that a new prefix be established? Why is it necessary that the courses be distinguished from those of other academic units?
- Describe the curricula with which the courses offered under the new prefix are to be associated.
- Describe the organizational unit that administers the courses listed under the new prefix.
- List the courses that will be offered under the new prefix.

The proposal should originate from the organizational unit that will administer the new prefix. It should be approved by appropriate unit and collegiate procedures, and should be signed by the head of the originating unit and by the supervising dean.

If the prefix will be associated with graduate courses, the proposal should be forwarded to the dean of the Division of Graduate Studies, who will review the proposal and send his or her recommendation to the Office of the Executive Vice President and Provost of the University. If it is anticipated that only undergraduate courses will be offered under the proposed prefix, the proposal may be sent directly to the Office of the Executive Vice President and Provost of the University.

The executive vice president and provost of the university will review the proposal and the recommendations of the graduate dean and/or supervising dean. Notice of approval or disapproval will be sent to the originating unit from the Office of the Executive Vice President and Provost of the University.

College/School/Division	**Technology and Innovation**		
Department/Unit	Engineering		
Prefix (3 letters)	SER	Prefix name	Software Engineering
Effective Term	Fall	Effective Year	2011

Curriculum Contact Person (Academic Department): Chell Roberts, Chair, Department of Engineering **Phone:** (480)727-2727

Approved by the Graduate College (for graduate-related prefixes only) Date

Signature – Executive Vice President and Provost of the University (**final approval**) Date

Processed by- University Registrar's Office Date

ARIZONA STATE UNIVERSITY

September 15, 2010

To: Keith Hjelmstad, Vice President and Dean, College of Technology and Innovation
From: Ron Askin, Director, School of Computing, Informatics and Decision Systems Engrg.

RE: Proposal for BS in Software Engineering

I have reviewed the proposal to establish a B.S. in Software Engineering degree at the Polytechnic campus. The School of Computing, Informatics and Decision Systems Engineering believes there is a growing and unmet societal need for software engineers. The proposed program provides students with an alternate means to acquire software engineering education that can help meet this need. As such, we have no objection to the establishment of this degree program. Additional comments are provided below.

- The School of Computing, Informatics and Decision Systems Engineering on the Tempe Campus offers a B.S. in Computer Science degree with a concentration in Software Engineering. The curriculum for this existing degree and the proposed Poly degree both include software development projects and team experiences. In addition, both have similar program outcomes due to the requirement of satisfying ABET's "a through k". However, our focus is on the underlying science and principles that allow the graduate to understand complex software systems and the broader issues involved in software system design and assurance. The Poly program would appear to put greater emphasis on team-based projects and the application-focused practice of software generation. Thus the programs provide different options for students and employers.
- Understanding the difference between the existing and proposed programs may be difficult for prospective students, parents and employers. As such, it is essential that the programs be co-marketed to incoming students and employers to avoid subsequent dissatisfaction on their part. We are committed to working with the Poly campus to develop appropriate marketing materials that highlight the strengths and goals of each program. Presentation of a unified message to the outside will be important for minimizing confusion.
- As the proposed program provides a difference in focus and location from the existing Tempe program it is expected to only have a small impact on enrollment in the BS in Computer Science program on the Tempe campus.
- Students would benefit from the option of taking courses, particularly upper division electives, from either program. Thus we would see it as a benefit to both programs to allow majors to enroll in courses at the other location as electives and to consider placing some courses on-line for convenience of the students. In particular, we have strengths in areas such as software assurance/security and distributed computing that could be of interest to Poly students and for which there is no need to duplicate efforts.

ASU ARIZONA STATE UNIVERSITY

POLYTECHNIC CAMPUS

October 5, 2010

MEMORANDUM

TO: Elizabeth D. Capaldi, Executive Vice President & University Provost

FR: Keith Hjelmstad, University Vice President and Dean
College of Technology & Innovation

RE: B.S. Software Engineering
Program Fee

Upon the advice of the University Provost's Academic Council, I am writing to confirm that is our intention to move forward with implementing the B.S. Software Engineering program even without the program fee.