

Statement of Purpose

Your responses to these questions are an important part of your application package, and provide the Committee with information that is vital to their evaluation of your application. Please answer each set of questions completely and concisely. Avoid answers that don't directly answer the specific question. Please observe all page limits, font sizes, and spacing specifications. Each part may be 1 page with a 2-limit page total.

PART 1: Experience and Background

Please answer the following. Limit your response to 1 page, 11-point font, single-spaced.

- Describe a challenge that you experienced on a software development team, and what the preferred outcome was, or what you would like to have changed.
- When reflecting on your experience, please relate it to one of the six practice areas below and be specific.
 1. Architecture & Design
 2. Software Construction
 3. Planning & Tracking
 4. Quality Management
 5. Requirements Management
 6. Risk Management

PART 2: Purpose, Goals, and Achievements

Please answer the following. Limit your response to 1 page, 11-point font, single-spaced

- Describe your short-term goals (2-4 years) and long-term goals (5-7 years) upon graduation from this academic program. Feel free to describe your goals in terms of any relevant roles, domains, organizational technologies, expertise, and so forth. Describe specifically how the program to which you are applying, and Carnegie Mellon University more generally, aligns with your professional objectives.
- **Optional:** Briefly list any awards or distinctions that you have earned (one sentence for each).
- **Optional:** Provide a list of references publications that you have authored or co-authored.
- **Optional:** Please include additional information about yourself that you would like this year's Admission Committee to know.

SIX KEY PRACTICE AREAS

Architecture & Design

Software Architecture is designed to allow designers to indicate the significance of architectural concerns in a project and how they intend to support these concerns. Architectural concerns cover the design, analysis, and achievement of systemic properties (e.g., response time, scalability, availability, etc.); serve to guide and constrain construction and analysis; incorporate commitments in legacy code, open source software, and commercial components; coordinate with external development efforts; and enable transition of the system to future development and maintenance teams.

Software Construction

This practice area concentrates on the construction and management of software artifacts produced to implement a project. Teams should show competence in: (a) managing code artifacts (e.g., version and configuration management), (b) developing policies for managing its construction (e.g., coding practices/standards, code-focused processes such as TDD if that is relevant, pair-programming or other agile development techniques), (c) integration and validation with architecture and requirements, (d) issue management, and (e) managing software deployment and delivery to clients. Teams should document plans and practices, track compliance, and maintain traceability between this and other related activities (e.g., architecture, quality).

Planning & Tracking

This practice area concentrates on the establishment of plans that define project activities and the understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan. Teams should show competence in: (a) developing strategic plans, (b) developing tactical plans, (c) tracking the work progress, and (d) updating the corresponding plans in response to deviations. Teams should document plans and practices, track compliance, and maintain traceability between this and other related activities (e.g., architecture, quality).

Quality Management

This practice area concentrates on quality management tasks and activities on the project. This should include establishing quality mechanisms consistent with the needs of the client and the project, and may include prevention as well as detection mechanisms. Teams should collect, and have available, basic quality plans and metrics consistent with progress on the project. For example, preconstruction teams/projects would have initial plans for what quality mechanisms and metrics are planned for the duration of the project, whereas during construction teams/projects could collect metrics on defects.

Requirements Management

This practice area concentrates on the requirements gathering, analysis, validation and verification activities needed to ensure that the software development team is building the right software. This includes identifying the business objectives that drive the software requirements, ensuring that all stakeholders and their needs are accounted for, and

documenting and tracing requirements to downstream activities where they are needed to make important design and quality assurance decisions. It is common to surface business and technical risks and obstacles during these activities, and to introduce requirements to manage those risks.

Risk Management

This practice area concentrates on risk management tasks and activities on the project. This should include basic project risk management such as: process/plan to address risk management on the team/project to enable identification of risk and mitigation strategy, risk assessment including probability/impact/severity/priority/exposure, and ongoing monitoring/tracking. Teams should document plans and practices, track that these are being followed, and maintain traceability between this and other related activities (e.g., construction, planning, requirements etc.).