INSY 560 Test 1 Objectives

The following are things you should be able to do or discuss on the first test. The test will probably consist of approximately six essay- or form-completion type questions. You will be expected to write approximately 1-2 pages for each answer. Your answers will require both direct writing, as well as diagramming of the information. The form-completion type questions would provide information and one or more forms, and you would have to fill out the forms based on the information provided.

General

- Diagram and/or explain the *5-level CMM* and the *3-level PSP* as described in our textbook. Give an overview of each of these hierarchies, explain their various details, and describe their usefulness and application to software engineering (SE).
- Describe / explain the *general approach* advocated by the PSP.
- Describe / explain in general what *scripts*, *forms*, *logs*, and *standards* are in the PSP, and what each is used for.
- Describe / explain in detail the various *scripts*, *forms*, *logs*, and *standards* which are used in each portion of the PSP, and be able to fill out each type of form or log based on your actual work or if given sample data.

Ch. 0: Preface

- Explain / discuss the issue of whether software (SW) development is an *engineering discipline or a craft*. With reference to the two articles, "Prospects for an Engineering Discipline of Software" (Shaw, 1990), and "Measuring and Managing Technological Change" (Bohn, 1994), describe and discuss the phases followed in moving from a craft to an engineering discipline, and explain the eight *stages of knowledge* and the *evolution of knowledge* and performance.

Ch. 1: The PSP Strategy

- Define and describe what is meant by a *software process*.
- Define and explain what is meant by *process maturity*.
- Diagram and explain the 5-level CMM and it's key process areas.
- Diagram and describe the 3-level PSP and it's key process areas.

Ch. 2: The Baseline Personal Process

- Describe how a *defined process* can be *helpful* in software development work.
- List and explain the basic *process elements of PSP0*.

- Describe the *basic workings of the time log* and what its usefulness is, and be able to properly fill out a time log based on your actual work time or if given sample data.
- Describe the *basic workings of the defect log* and what its usefulness is, and be able to properly fill out a defect log based on your actual work time or if given sample data.
- Describe the *basic workings of the PSP0 project plan summary* and what its usefulness is, and be able to properly fill out a PSP0 project plan summary based on your actual work time or if given sample data.

Ch. 3: Planning I: The Planning Process

- Describe / define what a *plan* is, and explain how planning is useful in software development work.
- Describe how the *contents of a software plan* are determined — what are the contents based on?
- List and describe the *four steps in planning* a software project, and explain how these steps fit in with the overall Personal Software Process.
- Diagram the *planning framework*.
- Name and describe the *six aspects of a high quality plan*.

Ch. 4: Planning II: Measuring Software Size

- Name the three *criteria* for software size measures.
- Define and describe how *correlation* (r), *variance explained* (r^2), and *alpha* (α) are used in helping us determine the usefulness of any software size measure (or any measure for that matter).
- Define and explain the terms *precision*, *accuracy*, and *reliability*, and discuss how they relate to software measurement and the activities which are performed in the PSP.
- Describe what a *LOC counting standard* is and what one should include, and explain why such a standard is extremely important with respect to (WRT) software size measurement. Be able to create your own LOC counting standard, justify your decisions in how to count LOC, and give examples.
- Explain what is meant by *logical*, *physical*, and *logical/physical LOC counting*.
- Discuss why it is important to have a *coding standard* in addition to a LOC counting standard. Explain what such a coding standard should contain. Be able to create your own coding standard, justify your decisions, and examples.
- Discuss the various issues which are important to take into consideration when using LOC as a measure.
- Describe and explain what a *process improvement proposal* (PIP) is and how it can be beneficial in the SW development process. How does the PIP enhance the PSP? What does its use cause to happen to your process?

Ch. 5: Planning III: Estimating Software Size — Estimating Methods, Proxies, and the PROBE Method

- List, describe, and explain the *criteria for a size-estimating method*.
- Name and briefly describe the *five estimating methods* discussed in the chapter: Delphi, Fuzzy Logic, Standard Components, Function Points, and Proxy-Based Estimating. Discuss advantages and disadvantages of each.
- Define and explain what a *proxy* is, and give examples WRT software size measurement.
- Name and explain the *criteria for a good proxy*.
- Diagram and briefly describe the *PROBE* method.
- Discuss the use of *object categories* in the PROBE method.
- Explain how to determine *object categories and subcategories / subranges* for use in PROBE. Be able to determine these categories / ranges from your own PSP data or from sample data.

Ch. 6: Planning IV: Resource & Schedule Estimating

- Describe in general what *resource planning* is all about.
- Diagram and explain the *development time planning process*.
- Describe the statistical technique of *regression*, and discuss in general and explain in detail how regression is used in software engineering planning. Give an example. Be able to work problems using regression to plan software development activities.
- Describe the difference between simple and *multiple regression*. Discuss why one would be used over the other.
- Describe what *schedule estimating* and setting is all about.
- Define the concept of *earned value* (EV), and explain how EV can be used in schedule estimating and tracking. Be able to work earned value calculations on your own PSP data or on sample data supplied to you.

Ch. 7: Measurement in the Software Process

- Discuss the reasons *why it is valuable to make measurements*.
- List and describe the *principal measurement categories*.
- Describe what the *GQM* framework is, and explain how it is helpful WRT measurement in the SW process, and particularly with the PSP.
- Describe what an *engineering notebook* is, explain how one might be used, and discuss the value of maintaining such notebooks. Keep a notebook of your own. Illustrate the use of such a notebook.

Ch. 8: Design & Code Reviews

- Discuss the *importance of design and code reviews*.
- Name and explain the *three types of reviews* discussed in the textbook.

- *Justify why time should be spent on reviews* in the SW development process. Discuss the tradeoffs that are realized.
- Describe the *single biggest problem with reviews*, and discuss how it can be overcome.
- Discuss the *efficiency of reviews* at various points throughout the SE process, and describe how one can decide at which points it is worth performing reviews.
- Name and discuss *principles* which should be followed in order to make reviews effective.
- Name and describe several *measures* which may be used to assess the *effectiveness of reviews*.
- Discuss how *checklists* can be used to facilitate the review process.
- Discuss the pros and cons of *reviewing before vs. after compiling*, and how to decide when to perform your reviews. Discuss how this may change over time and why.