

3.2 Project Steps and Workproducts

This is the third time for this class to be offered we may be modifying some of the project steps and workproducts as we go along. So this list is preliminary and subject to discussion and revision. The schedule is negotiable by group since different representations and toolsets may have different overheads for use.

- a) Consistent project specification – The first workproduct is your revision of the project specification you were given.
- b) Properties in English – The second workproduct is an English specification of the set of properties which will define “correctness” and/or reliability for your project. (Workproduct) Due – 9/25/2008
- c) Formalization of the properties in the specification logic which for the time being will be a propositional logic extended with temporal logic (LTL or CTL) including past time and metric logic where needed or JML (Workproduct) Due - 10/2/2008)
- d) Representation of the design for implementing the specification following the principles of design for verification and validation together with a document explaining how your design satisfies the principles of designing for verification. The specific work products are: (i) an architecture specification which can be an annotated diagram showing components and the relationships between components. (ii) definitions for each component either in English, as pseudo-code or as an annotated code framework. For example, if you will be coding in Java, each component may be stub for a Java class where the class data is defined and the method interfaces are defined. (iii) the relationships can be represented as a control flow diagram at the level of method calls or as a state machine. If these definitions are not clear, please send email or consult with the TA or the instructors. (Due 10/14/2008)
- e) If necessary, specify the execution environment for your system, The execution environment is the set of inputs the system will receive from the external world and the responses the external work will make to outputs provided by the system. Then use the architecture diagram and the system level properties you have defined in step c to define the environment in which each component executes and the properties (in your chosen temporal logic) for each component which are required for system level properties to hold. The process is to follow the execution of the system through the architecture diagram and observe the requirements which each invocation of a component requires of it. That is the required property. If this is not clear – see the TA or the instructor. The work product is the set of properties for each component in your chosen temporal logic. (Due 10/21/2008)
- f) Test plan – The work product is a test plan which is derived from the properties using a process which will be explained in lecture. This test plan is a document which defines a set of tests which will informally verify each property. (Due 10/28/2008)
- g) Completed code and informal tests for the system in the implementation language of choice. (Due 10/31/2008)

- h) Initial mapping of properties to verification methods. This step will need to be done in consultation with the TA and the instructors. This step should have been preceded by sufficient testing to enable application of formal methods.(Due 11/7/2008)
- i) Model checking for one or more properties for one or more components. (Workproduct in the form of an analysis of success or failure.) (Due 11/16/2008)
- j) Design and execution of proof process for one or more properties for at least one component of the system. (Workproduct is a report on success or failure.) (11/23/2008)
- k) Design, implementation and experimental evaluation of runtime monitors for selected properties.(Workproduct is a report on success or failure.) (11/30/2008)
- l) Final Report – A paper summarizing what was accomplished on the project and what you learned and enough documentation to enable me to follow it. This will be done by each student separately. (12/6/2008)

Project groups may propose alternative methods of verification such as use of symbolic execution or static analysis methods or incorporation of symbolic execution or static analysis methods in one or more of the verification steps.