

T. H. E. SOLUTION

Product Development Consulting

UNDERWRITERS LABORATORIES, SOFTWARE AND YOU

INTRODUCTION

UL has recognized software as a crucial element in product safety and has also recognized that the traditional UL approach to investigation is not practical for coping with this area. Instead, the investigations will now look at client produced artifacts from the design and deployment of the product. This includes three phases of activity by UL. The first phase reviews the design using a documentation review and an on-site visit. The second phase includes auditing visits. The principal area of interest during these visits is the safety aspects of the system development process. These visits are intended to eliminate the need for new phase one reviews for every new release of a product. The third phase is composed of unannounced factory floor inspections. Here the concern is the deployment aspect of the product cycle.

Specific investigation programs will be designed for each industry that produces safety critical products. Product areas that have been identified include programmable electrical medical systems, programmable industrial control systems, fire alarm and control systems and electric vehicle and automotive control system. In each case UL expects to work with industry groups to put the revised process into place.

The primary question you must answer is not *will* your company comply with the new UL processes but *how* will you comply and *when*. Before answering these questions let's look at some important background information.

BACKGROUND

At this point UL has provided only brief descriptions on what they will require. For example the UL standard for software safety, UL 1998, is very sketchy beyond the area of hardware fault detection. Basically, UL expects to see processes and the results of processes, artifacts, which show a strong concern with product safety. Based on communications to date, UL expects a rigorous approach to product development and deployment. Topics such as hazard analysis, specifications, standards and practices, formalized testing programs and cradle to grave configuration management have been included in UL presentations and are briefly touched in UL 1998. The definition of "strong concern" regarding product safety will probably vary on an industry basis. How each industry is rated will probably be based on the perceived safety risk to the public. Where human life can be dramatically impacted, you can expect to see "best effort" requirements.

What is best effort? Does this mean perfect software? With current knowledge and methods, the goal of producing perfect software is unrealistic. Experts cannot even agree on a single definitive approach to reaching this goal. Use of reliability models to predict zero or extremely low probability of failure is disputed on both theoretical and practical levels. Total fault tolerance is not commercially viable since it is highly dependent on physical redundancy. However, there does appear to be a reasonable consensus that more "mature" organizations produce better (and safer) software. Different people and organizations use slightly different means of defining and determining maturity. A useful way to look at this is to think in terms of a company's product

development and deployment practice. The maturity of this practice is a measure of how well the organization brings together people, process and tools. A useful framework for talking about practice maturity levels is a simple segmentation into three groupings: basic, good and best. A summary description of each of these levels for the three primary technical disciplines in software production is provided in the following tables.

Discipline	State of Practice
Project management	<ul style="list-style-type: none"> process phases and work products defined formal review between phases organization/role definitions based on multi-functional teams project metrics selected but little historical data project status parameters loosely monitored over commitment of resources recognized and corrected basic group work tools in place
Software engineering	<ul style="list-style-type: none"> process steps and work products defined automated configuration management primary metrics in place but little historical data peer reviews at least between steps system integration testing before release to evaluation source closely scrutinized for language violation practices normally followed
Quality assurance	<ul style="list-style-type: none"> inspection/testing aimed at unit, sub-system and system level formal verification in system level testing automated defect tracking and some automation of testing some QA personnel participation in early work product review

Basic Level of Practice Maturity

Discipline	State of Practice
Project management	<ul style="list-style-type: none"> regularly learn from past tight integration with marketing, operations AND customer roles well understood and comfortable planning complete and sanity checked with historical data project status parameters closely monitored over commitment of resources not allowed
Software engineering	<ul style="list-style-type: none"> formal inspections of all work products high automation of design and documentation source scrutinized for any sign of potential problems practices consistently followed all participants trained in process and tools
Quality assurance	<ul style="list-style-type: none"> formal reliability engineering approach in place most testing automated heavy QA personnel participation throughout

Good Level of Practice Maturity

Discipline	State of Practice
Project management	majority of work force is experienced in process strong emphasis on risk management
Software engineering	majority of work force is experienced in process and tools good domain knowledge low performers have been eliminated heavy reuse of existing proven correct code
Quality assurance	defect cause information used in root-cause analysis

Best Level of Practice Maturity

Current evaluations indicate that over 70% of development companies do not even meet the basic level of maturity. Even the best companies meet the best level of maturity only rarely. It is highly unlikely that any development in a new product area can ever meet the best level. What does this mean? Benchmarks for productivity, quality and development costs for each of the maturity levels are summarized below:

Level of Practice Maturity	Productivity (LOC/man-month)	Direct Development Cost (\$/LOC)	Quality to Customer (failures/KLOC)
basic	250 - 600	18.30	4
good	700 - 1500	6.63	.1
best	> 2000	3.28	.001

LOC - lines of code

KLOC - thousand lines of code

failures - deviations from requirement/specification during execution

You will notice that even with the highest level of practice maturity, you do not produce perfect software. You will produce very good software at a very competitive cost, but it will not be perfect. Another interesting observation is that productivity rises and development cost falls with rising quality. If you consider the entire development - deployment cycle, doing it right does pay off.

OK, you cannot produce perfect software, at least not yet. You may want to aim for it but the best you can expect with current knowledge is a good level of quality, i.e. a failure density of approximately .1 failure per thousand lines of code. Is this best effort? Can you intensify your focus on safety critical factors to prevent any (or at least most) of the failures from being safety critical? The answer to this question, in theory, is yes. The discipline of hazard analysis is intended to provide the means to focus on safety areas. Hazard analysis is an umbrella container for a substantial body of knowledge, techniques and methods that can be brought to bear on safety critical problems. It can be sub-divided into two different approaches, inductive and deductive.

The inductive approach works from the detailed cause to the general effect. Many methods of inductive analysis exist. Examples include failure mode and effect analysis (FMEA), failure mode effect and criticality analysis (FMECA) and fault hazard analysis (FHA). Though these methods are useful, each suffers from the requirement for detailed design knowledge. In addition, each typically requires the manipulation of a large amount of data. As a result, these methods are generally not useful in the early phases of product development.

The deductive approach works from the general effect to the detailed cause. The best-known deductive method is fault tree analysis. Since this method works from the general effect, (i.e. the failures to be avoided) to their causes, it can be used early in the development process and it can be revised as the depth of product knowledge increases.

From all of this, we can finally answer the primary question of interest: What is UL compliance going to demand from my company? For products that have a significant impact on human safety, the UL will require a "best effort" in the product's development and deployment. In terms of the development practice this will require a good level of practice maturity and the use of hazard analysis methods to focus on safety critical factors.

A COST EFFECTIVE SOLUTION

Now we can finally address the initial question: How and when should my company comply with the UL software processes? There are at least two generic means of how. The first is to comply in form. In other words learn to produce the artifacts that satisfy UL requirements. The concern is not with higher quality or better safety per se but with meeting UL standards. This compliance in form is not an atypical approach. Evidence of this are the reported software development failures of ISO 9001 compliant developers. There is at least one overriding business reason not to take this approach., companies that actually improve their development practice will eventually have better products and bottom lines. Compliance by form will not keep your company competitive.

The second approach is to comply by practice. This requires improvement of your development practice to a good level of practice maturity and augmentation of this with specific cost-effective techniques for hazard analysis. This approach does not come free, but the payoff includes improved competitiveness not just UL compliance. Investment paybacks as high as ten to one have occurred in the process.

Most companies need to make the change while minimizing interruption to ongoing projects and minimizing investment costs. Given this criterion, implementation of a good practice level should be approached using continuous change. This is the gradual improvement of the practice with the active participation of the practitioners. This approach was used by Japan following World War II to revitalize its industry with marked results. A side benefit of this approach is that it provides a mechanism for continual improvement. This is important since competitive productivity, quality and safety are not static targets but change as the state of the practice changes. Conventional wisdom is that most organizations, i.e. those that do not currently meet the basic level, should be capable of meeting the good level in two years. This probably answers the question of when your company should start working on compliance, Now!

Selecting the means to focus the development practice on safety critical factors is not only important to assure UL compliance but will also affect the cost of compliance. One observation drives the selection: The earlier you can focus on the principal safety factors the better you will be

in both technical and business terms. Based on this, an outline of the suggested augmentation technique for life/safety critical applications is summarized in the following table. In the table, specific actions are mapped to the development phase of the project. This approach has been successfully applied to safety/life critical medical and non-medical products.

Development Phase	Action
System conception	Establish product failures of concern
System definition and design step of technical implementation	Determine cause and effect paths that lead to failures using fault tree analysis Formulate means to block each path OR to detect and assume a "failsafe" posture prior to failure Verify correctness of hazard analysis
Technical implementation	Implement carefully
System evaluation	Verify that blocks and/or detect with failsafe works * exhaustive testing if possible * statistically meaningful testing otherwise

Several of the actions described require a little more explanation. The first is the concept of blocking a failure path or detecting it and assume a failsafe posture. A "block" is simply a mechanism that prevents a hazard producing cause from occurring. Traditionally these have been mechanical or electrical mechanical mechanism. Software can often aid in blocking hazard paths but seldom can it do the whole job. Detection and assuming a failsafe position is also conceptually easy. When a failure mode is detected the device assumes a state from which it can cause little or no harm. This also normally requires more then just software to implement.

The next action that requires more definition is that of "implement carefully". Having a good level of practice already means that you are using rigor in both your technical approaches and your management. In the case of life/safety critical items this is often still not sufficient. For these items additional safeguards must be established. One such technique is the use of formal tractability. Please note that this is not advocating the use of formal tractability for everything in the development. As evidenced in DOD developments this places an onerous burden on the developer that often results in no appreciable improvement in quality. Only the specific items in the development that deal with the hazard paths should be dealt with in this manner.

Finally, the evaluation phase needs a little more explanation. It is usually impossible to conduct exhaustive testing for complex products. However, when just the failure paths are considered this may not be true. If it is at all possible to do exhaustive testing, then do it. If not, the same formal reliability engineering approach used in general should be used for testing the effectiveness of the blocks etc. Testing of these measures should take place both as part of normal operation test and as a separate round of stress testing. Whether exhaustive or statistical, this will usually require that the testing be automated.

This safety factor augmentation should be included as another aspect of building practice maturity. Because of its critical nature this augmentation should be in place well before meeting the good level of maturity.

CONCLUSION

UL's changing attitude on software may well challenge your company. However, as indicated in this article you can not only meet this challenge but also prosper if you make the right changes. With the right combination of people, process and tools your company can be a winner in this increasingly challenging business environment.