

UNIVERSITY of the
WESTERN CAPE

UNIVERSITY OF WESTERN CAPE (UWC)

SYSTEMS DEVELOPMENT LIFECYCLE (SDLC) POLICY (ICT POL!);')

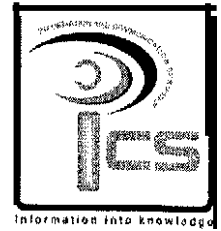
DATE OF LAST APPROVAL: C2011/02

Approved by Council, 9 July 2011 (C2011/02)

SECRETARIAT
Private Bag x17, Bellville 7535,
South Africa

2027 / 2146
3443
w@uwc.ac.za
, <10 _ zt.1

A place of quality,
a place to grow, from hope
to action through knowledge



Document Information

Document Type	ICT POLICY
Document Title	University of the Western Cape SDLC Policy
Document Number	UWC-ICS-018
Issuing office	Information and Communication Services

Document Control

Preparation by:	Anver Natha
Contribution by:	Shirleen Langenhoven
Contribution by:	Enter the name of the contributor
Contribution by:	Enter the name of the contributor
Approval by:	EAISC, UWC COUNCIL
Distribution List:	EAISC, UWC COUNCIL, ICS Department
Implementation Date:	Enter the date the policy/procedure comes into effect

Version Control

Version	Date (dd, mm-yyyy)	Contributor	Change Description
Version 2.1	18-10-2010	Anver Natha Shirleen Langenhoven	Initial Submission
Version 2.2	20-04-2011	Anver Natha	Reformat, separate policy from procedure

This policy /procedure will be reviewed annually for any necessitated changes

Introduction

The University of the Western Cape (UWC) has adopted a mixed approach to acquiring the software that it needs for its administrative systems. Some parts are based on non-proprietary, others on proprietary products. This runs the risk of eclecticism and lack of alignment, with significant implications for the skills set needed in ICS and for costs. This document therefore describes the SDLC framework followed in acquiring and implementing software solutions for administrative systems and acts as a guideline for ICS decision making processes.

Purpose

This Systems Development Lifecycle (SDLC) methodology defines the procedures governing the development, maintenance and retirement of Information Technology systems applied in support of the University business processes.

Scope

This SDLC methodology governs

- The development or acquisition of generic business applications (incorporating their supporting infrastructure)
- The maintenance of existing generic applications or their supporting infrastructure
- The acquisition or upgrade of all University Information, Technology or **Communication assets.**
-

These lifecycle procedures are not applicable to single usage (commodity/desktop) software or hardware unless such hardware or software directly impacts the integrity of the University IT generic applications applied in support of the business processes.

Assumptions & Constraints

The following assumptions were made in creating this methodology:

- The IBM Rational Unified Process (RUP) for software engineering is the de facto methodology incorporated by the ICS department in the development and support of University business applications.
- The UWC SDLC description forms the basis for the framework and is an amalgam of **various methodologies currently in use at UWC.**

The methodology is not constrained by any known circumstance.

Abbreviations & References

Abbreviations

SDLC	Systems Development Lifecycle
RUP	Rational Unified Process
COTS	Custom off the shelf
ICS	Information and Communication Services Department

References

IBM Rational Unified Process - software engineering methodology.

Overview

This System Development Lifecycle methodology presents **five** SDLC Frameworks based on the circumstance of the application system under consideration - these SDLC Frameworks form the foundation of procedures to be followed in the application development effort.

These five elements have been adapted from the IBM RUP methodology for UWC purposes. The SDLC frameworks and where possible the terminology used in the SDLC Framework are aligned with the RUP Methodology. Accordingly the application development projects should reference the RUP Methodology when applying the framework.

Initial sections in this methodology assist in interpreting the SDLC Frameworks presented.

The appendix provides a glossary of terms used within the frameworks.

Interpreting the University IT SDLC

University SDLC & RUP Methodology

SDLC Frameworks

Just as each business application system has unique attributes - so the development activities and associated deliverables in creating & enhancing each diverse system are equally unique. In an attempt to avoid creating a common SDLC framework applicable to all systems, the following strategy is adopted.

- A limited set of SDLC frameworks is presented (5-outlined below). These incorporate the minimum set of activities and deliverables required to satisfy University policies & procedures for the system development effort.
- These are complemented by a guide as to which SDLC framework is applicable, based on the type and apparent risks of the system development effort.
- A deliverable in the early milestone of the SDLC that describes the framework for the development process (a "Development Case" (RUP)) specific to the application system under consideration - effectively providing a unique SLDC for the system.

1. SDLC for Projects

There are two "Projects" SDLC frameworks presented in this methodology to provide for the traditional University IT Project -

- **University Applications:** in which the application system is created "in-house" such that the systems functionality is, mostly, unique to the University business, and-

2. SDLC for Projects

- **COTS Applications:** in which the application system is, mostly, evolved from commercial off the shelf (COTS) software (i.e. a "package" solution).

3. SDLC for Enhancements

- There is a single common "Enhancements" SDLC framework presented for either University or COTS applications which reflects the minimum activities & deliverables when enhancing an existing system.

4. SDLC for Maintenance

- There is a single common "Maintenance" SDLC framework presented to cater for "patch" maintenance of an existing application system.

5. SDLC for Infrastructure Assets

- The SDLC framework for infrastructure assets specifically addresses the activities and deliverables required when adjustments to the supporting infrastructure (hardware & generic software) are under consideration - even though no enhancements may be intended for the business application's functionality (that is - no code or data structure changes are involved).

SDLC Framework

Each of the SDLC frameworks is presented in this methodology as a single page, incorporating a cross-reference (Framework) of the SDLC milestones (7) and their artifacts, essential activities and milestone-end review points. This Framework page, together with the milestone-end list of checkpoints, provides a quick reference to all the minimum requirements of the SDLC.

RUP Methodology & SDLC References

The SDLC frameworks are based, for the most part, on activities and artifacts best described by the Rational Unified Process (RUP) Methodology.

RUP is not a single concrete prescriptive process, but rather an adaptable process framework, intended to be tailored by the development organizations and software project teams that select the elements of the process that are appropriate for their needs.

The RUP articulates - six *best practices* for modern software engineering:

- **Develop iteratively, with risk as the primary iteration driver**
Manage requirements
Employ a component-based architecture
- **Model software visually**

- **Continuously verify quality**
Control changes

From: Wikipedia: IBM Rational Unified Process

Where there is significant detail referenced in the RUP Methodology - the SDLC Framework's artifact or activity is suffixed with "(RUP)" (e.g. Business Case(RUP)), which can be explicitly searched for in the RUP Methodology. References to activities and artifacts that are, for the most part, specific to the University are expanded on in the Glossary sections of this methodology.

SDLC- Milestones

There are seven milestones within each of the SDLC frameworks. These are outlined below. However, only the milestones pertaining to software engineering practices are referenced in the RUP Methodology (Inception, Foundation Installation, Construction and Implementation) - the remaining milestones are formulated specifically for the University IT SDLC.

I. Initiation Milestone

The Initiation Milestone establishes the application effort as a University project.

II. Inception Milestone

The objective of the Inception Milestone is to achieve concurrence among all stakeholders on the objectives for the project and establishes that the project is worth doing and that one or more feasible candidate solutions exist.

III. Foundation Installation Milestone

The objectives of the Foundation Installation Milestone are to baseline the architecture of the system and to provide a stable basis for the bulk of the design and implementation effort in the construction milestone.

IV. Construction Milestone

The Construction Milestone clarifies the remaining requirements and completes the development of the system based upon the baseline architecture.

V. Implementation Milestone

The objective of the Implementation Milestone is to ensure that software is available for its end users.

VI. Production Milestone

During the Production Milestone the system is kept useful and productive after it has been deployed to the user community.

VII. Retirement Milestone

The Retirement Milestone is applied when the system is to be decommissioned.

SDLC- Deliverables

The "Glossary of Artifacts" section overviews the purpose of each artifact - however, further details (if needed) may be sought from the RUP Methodology or associated documentation templates.

Milestone-end Deliverables

The Milestone-end Artifacts (as well as their "Associated Artifacts") - as listed in the "SDLC Framework's" - are the minimum artifacts that are to be produced before commencing with the next milestone of the project.

If additional artifacts are defined for the milestone (as specified in the "Development Case") these should also be produced as Milestone-end Artifacts.

Associated Deliverables

Associated Artifacts (as well as their primary "Milestone-end Artifacts") - as listed in the "SDLC Frameworks" - are the minimum artifacts that are to be produced before commencing with the next milestone of the project.

These Associated Artifacts should be referenced in the primary "Milestone-end artifact" and be available on request during "Milestone-end Review Points".

Deliverable Templates

Documentation templates associated with an artifact are available. They have been cloned from the RUP Methodology and formatted in accordance with the University standard.

SDLC- Essential Activities

The section "**Glossary of Essential Activities**" overviews the purpose of each activity - however, further details may be referenced directly from the RUP Methodology.

The Essential Activities listed in the "SDLC Frameworks" identify those activities that are pertinent to the milestone of the project. However, during the software engineering milestones of the project (Inception, Foundation Installation, Construction and Implementation) the RUP Methodology should guide the process.

SDLC- Milestone-end Review

Review Points

Formal reviews of the activities and milestone-end artifact deliverables are required before commencing with the next milestone of the project. In certain instances these require formal authorization (signatures) from the reviewer.

The degree of review required is detailed in the associated "Milestone-end Check-Points" appendix section.

Reviewer Roles

The specific roles and responsibilities within the project as pertains to the review points are detailed in the artifact "Software Development Plan" - however, generic roles are defined below as being pertinent to the review of the Milestone-end activities and artifact deliverables. Milestone-end reviews are necessary for quality management and to ensure that the deliverables are aligned with the Business expectation and requirements

IT Quality Assuror

The role of the IT Quality Assuror (as listed in the "SDLC Framework's") at Milestone-end review is:

- To ensure, by review, that the subject matter of the artifacts are appropriate before they are presented to the Key Stakeholders or/and (project) Sponsor.
- To ensure, by review, the quality of the development process through conformance to the University IT policies & procedures.
- To ensure, by review of the Milestone-end Check-Points, that the required activities and artifact deliverables of the milestone have been achieved.

The role of the IT Quality Assurer (QA) can be fulfilled by the Governance Office, Project Office or a designated person within a Project.

The assignment of the IT QA responsibility should be done in consultation with the Manager: Projects.

For fully outsourced projects the assignment of the Quality Assurer should be done in consultation with the Quality Manager of UWC, in conjunction with the Manager: Projects

Milestone-end checkpoint reviews and sign-off for the University Projects and COTS Frameworks are done by the Project Office and UWC Quality Manager (where applicable), to ensure that duties are segregated for the SDLC QA.

Key Stakeholders

The role of the Key Stakeholders at the Institution is to review the Milestone-end to ensure that the solution being produced realizes the expected benefits to the business and satisfies the requirements as defined.

Sponsor

The role of the Sponsor at Milestone-end review is to assess that the business benefits are realized and authorize the subsequent milestone of the project.

Milestone-end Check-Points

Milestone-end Check-Points, specific to each milestone of the SDLC, are located as appendices and should be read in conjunction with each of the SDLC Framework pages.

The appropriate Milestone-end pages should be extracted and completed by the IT Quality Assuror (University Projects and COTS Frameworks should be done by the Project Office) at each **Milestone-end to ensure that the activities and artifact deliverables have been produced in accordance with University IT's policies & procedures and the defined application development SDLC process.**

Note: Check-Points for the Production Milestone of an application are incorporated within the appropriate SDLC framework for application maintenance - as such there are no "Check-Points" defined for the Production Milestone of the SDLC.

Selecting the University SDLC Framework

Determining which SDLC framework is applicable to the application effort should be done during the **Initiation Milestone** of the project and in consultation with the Manager: IT Projects and/or IT Governance representative.

Considerations in selecting the appropriate SDLC Framework are that:

- **Project SDLC's** are synonymous with a formally registered University IT project.
- **Enhancement SDLC's** are synonymous with changes to existing applications with no apparent risks.
- **Maintenance SDLC's** are synonymous with "patches" or bug fixing to existing applications requiring urgent attention and are expected to incur minimal cost of effort with no risk to the integrity of the application.
- **Infrastructure SDLC's** are synonymous with adjustments to the business application supporting infrastructure - focusing on ensuring the integrity of the impacted application during the infrastructure update process.

Note: Bug fixing refer to correcting coding errors that occur in both project and production modes. Production bug fixing is classified as a project task once the fix will take longer than 40 hours to resolve.