ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

****

NHẬP MÔN CÔNG NGHỆ

PHẦN MỀM

Phương pháp phát triển phần mềm - RUP

GVHD: Nguyễn Công Hoan

Tên Nhóm: T2DV

Tên thành viên:

* Mai Thụy Ánh Tuyết - 16521409
* Nguyễn Đức Tùng - 16521396
* Trần Ngọc Phương Duyên – 16520313
* Nguyễn Thị Thu Việt - 16521434

TP HCM, Ngày 06 tháng 05 năm 2018

Bảng phân công công việc

|  |  |  |  |
| --- | --- | --- | --- |
| Stt | Tên công việc | Người thực hiện | Thời gian nộp bài |
| 1 | Phương pháp phát triển phần mềm-RUP | Phương Duyên | 07/05/2018 |
| 2 | Lịch sử ra đời của RUP | Phương Duyên | 05/05/2018 |
| 3 | Giới thiệu về phương pháp phát triển phần mêm – RUP | Phương Duyên | 15/5/2018 |
| 4 | Các nguyên tắc của RUP | Phương Duyên | 15/5/2018 |
| 5 | Lợi ích, hạn chế của RUP | Thu Việt | 05/05/2018 |
| 6 | Điều kiện áp dụng | Ánh Tuyết | 05/05/2018 |
| 7 | Quy trình của RUP | Đức Tùng | 04/05/2018 |
| 8 | Các công cụ của RUP | Phương Duyên | 05/05/2018 |
| 9 | Biểu mẫu áp dụng | Thu Việt | 05/05/2018 |
| 10 | Tổng hợp tài liệu, viết báo cáo, làm slide báo cáo | Ánh Tuyết  Phương Duyên | 07/05/2018 |

Mục lục

[1. Tổng quan về phương pháp phát triển phần mềm – RUP 7](#_Toc514157327)

[2. Lịch sử ra đời quy trình RUP 8](#_Toc514157330)

[3. Giới thiệu về phương pháp phát triển phần mềm – RUP 9](#_Toc514157331)

[4. Các nguyên tắc chính của RUP 11](#_Toc514157353)

[5. Quy trình phát triển phần mềm RUP 11](#_Toc514157360)

[5.1. Khởi đầu (Inception) 12](#_Toc514157361)

[5.2. Dự thảo chi tiết (Elaboration) 12](#_Toc514157362)

[5.3. Thực hiện xây dựng 12](#_Toc514157363)

[5.4. Chuyển giao 13](#_Toc514157364)

[6. Điều kiện áp dụng, các công cụ, biểu mẫu áp dụng của RUP 13](#_Toc514157365)

[6.1. Điều kiện áp dụng 13](#_Toc514157366)

[6.2. Các công cụ của RUP 15](#_Toc514157373)

[6.3. Biểu mẫu áp dụng 16](#_Toc514157379)

[7. Lợi ích và hạn chế của RUP 18](#_Toc514157386)

[7.1. Lợi ích 18](#_Toc514157387)

[7.2. Hạn chế 19](#_Toc514157388)

1. **Tổng quan về phương pháp phát triển phần mềm – RUP**

Trong những năm qua, việc xây dựng và triển khai các chương trình phần mềm đã góp phần phục vụ ngày càng tốt hơn cho công tác quản lý và điều hành của nhiều doanh nghiệp. Tuy nhiên, cũng không thể tránh khỏi những sai sót làm ảnh hưởng không nhỏ đến hiểu quả công việc của cán bộ quản lý, ảnh hưởng đến tiến độ phát triển, triển khai và bảo trì chương trình của cán bộ Tin học, trong đó một nguyên nhân nổi bật đáng chú ý là chưa thực sự áp dụng một phương pháp luận, một quy trình chuẩn được công nhận trong quá trình phân tích thiết kế, phát triển, thử nghiệm, triển khai chương trình dẫn tới chất lượng của chương trình tại thời điểm tung ra triển khai thử nghiệm là hết sức thấp; nhiều lỗi không được phát hiện sớm; cách tiếp cận phát triển ứng dụng không dựa theo công nghệ hướng đối tượng nên khi có thay đổi chính sách nghiệp vụ dẫn tới ứng dụng phải đắp thêm các chức năng mới nhưng hết sức chắp vá.

Thế nên năm 2002, một bộ phận của IBM – công ty Rational Software đã tạo ra RUP – một quy trình vòng lặp phát triển phần mềm. Đây là quy trình phân tích, thiết kế, phát triển, thử nghiệm và triển khai một hệ thống phần mềm đã được hầu hết các hãng phần mềm trên thế giới áp dụng thành công trong các dự án của mình.

1. **Lịch sử ra đời quy trình RUP**

Bắt nguồn từ mô hình xoắn ốc (spiral model) của Barry Boehm. Rational Approach được phát triển tại Rational Software trong những năm 1980 và 1990.

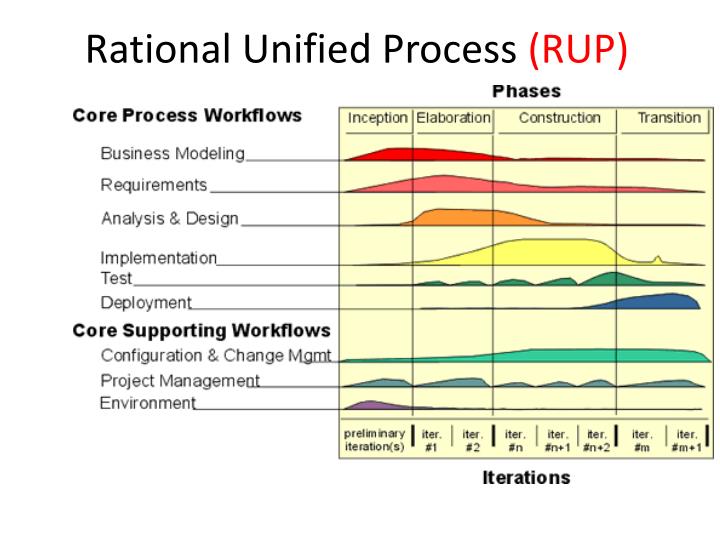
Trong năm 1995 Rational Software mua lại công ty Objectory AB. RUP là kết quả của việc trộn Rational Approach và quy trình Objectory được phát triển bởi nhà sáng lập Objectory AB là Ivar Jacobson, Objectory là một hệ phương pháp luận hướng đối tượng được mở rộng từ Ericsson Approach một ngôn ngữ mô hình hóa được phát triển bởi Ericsson.

Các kết quả đầu tiên của sự kết hợp trên được biết tới là Rational Objectory Process, RUP được thiết kế theo quy trình Objectory nhưng phù hợp với công cụ Rational Rose. Sau khi mục tiêu được hoàn thành thì được đổi tên thành Rational Unified Process, phiên bản đầu tiên là 5.0 được phát hành năm 1998, kiến trúc sư trưởng là Philippe Kruchten. Phiên bản cuối cùng là RUP 7.0 được phát hành là một phần của IBM Rational.

Động cơ phát triển RUP là tìm ra nguyên nhân của các lỗi phần mềm và tìm cách khắc phục chúng thông qua một quy trình.

1. **Giới thiệu về phương pháp phát triển phần mềm – RUP**

* RUP không phải là một quy trình bó hẹp cụ thể đơn nhất nhưng là một nền tảng quy trình thích ứng với sự phát triển các tổ chức và các nhóm dự án phần mềm, tất cả các yếu tố cần thiết của quy trình để phù hợp với nhu cầu, quy mô của công ty, dự án và sản phẩm.
* RUP là một liên kết các kiến thức cơ bản với các Artifact và mô tả chi tiết các loại activity khác nhau. RUP được chứa bên trong sản phẩm IBM Rational Method Composer (RMC) cho phép tối ưu tiến trình.
* Unified Process được thiết kế từ đặc điểm chung, quy trình phạm vi rộng lớn và RUP là một mô tả chi tiết cụ thể
* RUP hỗ trợ các hoạt động giữa các nhóm, phân chia công việc cho từng thành viên trong nhóm, trong từng giai đoạn khác nhau của quá trình phát triển phần mềm.
* RUP sử dụng hệ thống ký hiệu trực quan của UML và RUP được phát triển song song với UML.
* RUP là kết quả của nhiều “best pratcices”, được hỗ trợ nhiều công cụ phát triển phần mềm.
* RUP là một sản phẩm tiến trình có thể tùy biến.



* Cấu trúc của quy trình RUP được thể hiện theo hai chiều:

+ Trục hoành biểu diễn khía cạnh động của quy trình, bao gồm thời gian và các khía cạnh chu kỳ sống của quy trình

+ Trục tung biểu diễn khía cạnh tĩnh của quy trình, bao gồm các luồn công việc.

* Các giai đoạn công việc của RUP bao gồm:

+ Mô hình hóa nghiệp vụ (business modeling): mô tả cấu trúc và quy trình nghiệp vụ.

+ Xác định yêu cầu (requirement): mô tả kiến trúc hệ thống thông qua các sơ đồ phân tích thiết kế.

+ Phân tích và thiết kế (analysis & design): mô tả kiến trúc hệ thống thông qua các sơ đồ phân tích thiết kế.

+ Lập trình (Implement): thực hiện các việc xây dựng chương trình bằng ngôn ngữ lập trình.

+ Thử nghiệm (Test): mô tả các tình huống và kịch bản thử nghiệm, tiến hành thử nghiệm hệ thống phần mềm.

+ Triển khai (Deployment): đưa hệ thống phần mềm vào sử dụng.

+ Quản trị cấu hình và quản trị thay đổi (Configuration & Change Management): kiểm soát các thay đổi và duy trì sự hợp nhất của các thành phần dự án.

+ Quản trị dự án: quản lý toàn bộ quá trình làm việc của dự án.

+ Đảm bảo môi trường: đảm bảo các hạ tầng cần thiết để có thể phát triển được hệ thống.

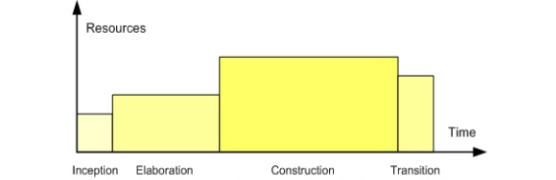
1. **Các nguyên tắc chính của RUP**

* Tuân thủ tiến trình (Adapt the process)
* Cân bằng các ưu tiên của stakeholder (Balance stackeholder priorities). Stakeholder là những người đưa ra những ý kiến có ảnh hưởng lớn tới tư tưởng thiết kế hệ thống
* Cộng tác giữa các nhóm (Collaborate across teams)
* Thể hiện kết quả theo quy trình lặp (Demonstrate value iteratively). Kết thúc mỗi quy trình cần theo dõi kết quả và kiểm tra chặt chẽ.
* Nâng cao mức độ trừu tượng (Elevate the level ò abstravtion)
* Tập trung theo dõi chất lượng liên tục (Focus continuously on quality)

1. **Quy trình phát triển phần mềm RUP**

Qui trình bao gồm bốn giai đoạn chính và đan xen nhiều dòng hoạt động (activity flow) như: Mô hình hoá nghiệp vụ, phân tích yêu cầu, phân tích và thiết kế, cài đặt, thử nghiệm triển khai, …Mỗi giai đoạn được hình thành từ những bước lặp (iteration).

Mỗi vòng đời phần mềm được chia thành nhiều vòng (cycles), mỗi vòng (cycle) làm việc trên một phiên bản mới của sản phẩm. RUP chia 1 vòng phát triền (development cycle) thành 4 giai đoạn (phase) liên tiếp: Inception phase, Elaboration phase, Construction phase, Transition phase.

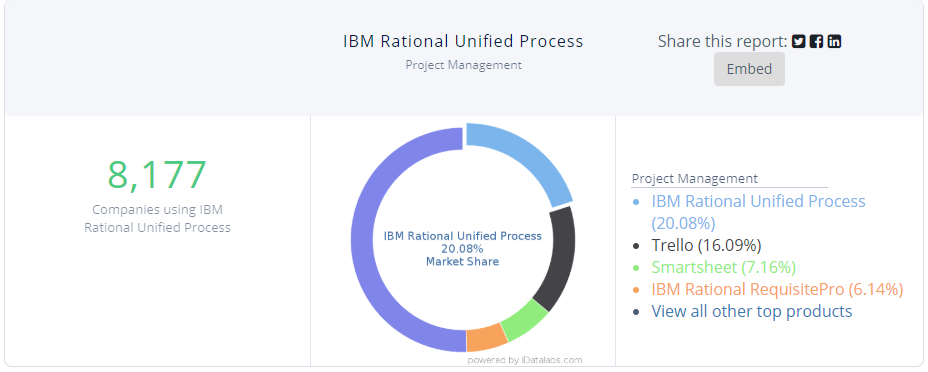


* 1. **Khởi đầu (Inception)**
* Giai đoạn thu thập thông tin (thực tế thiết kế và sản xuất) nhằm đặt ra mục đích và tầm mức của Dự án phần mềm:
* Phạm vi dự án, yêu cầu người dùng và ràng buộc
* Yêu cầu nghiệp vụ, rủi ro, kế hoạch dự án (phân công, chi phí)
* Phác thảo kiến trúc (chi phí, lịch, tài nguyên)
* Cấu hình môi trường làm việc, công cụ
  1. **Dự thảo chi tiết (Elaboration)**
* Phân tích, đánh giá các thông tin thu thập được nhằm xác định cụ thể, chính thức:
* Tầm mức dự án (project’s scope), các yêu cầu của dự án (project’s requirements), các điều kiện để dự án được xem là hoàn thành (project’s acceptance criteria).
* Các tính năng của dự án (project’s features), tính năng nào quan trọng (critical criteria), những rủi ro, mạo hiểm (potential risk)
* Chi tiết thực tế sản xuất (business specification) được dẫn đến xây dựng chi tiết về thiết kế phần mềm (design specification), xây dựng dự thảo cho kiến trúc phần mềm (software architecture), chọn thành phần (component).
* Những công việc hỗ trợ: thiết lập mạng, phần cứng, phần mềm, chuẩn bị quy trình, công cụ (CASE)
  1. **Thực hiện xây dựng**
* Chủ yếu là bàn về kiến trúc trúc phần mềm
* Quản lý tiến trình tạo sản phẩm đảm bảo năng suất, chất lượng. Quy trình thực hiện như tạo use case diagram, cài đặt môi trường phần mềm.
* Kế hoạch triển khai ứng dụng, người dùng
  1. **Chuyển giao**
* Kiểm tra sản phẩm
* Xuất xưởng, thu thập phản hồi
* Hỗ trợ hiệu chỉnh
* Kế hoạch cải tiến
* Phân phối

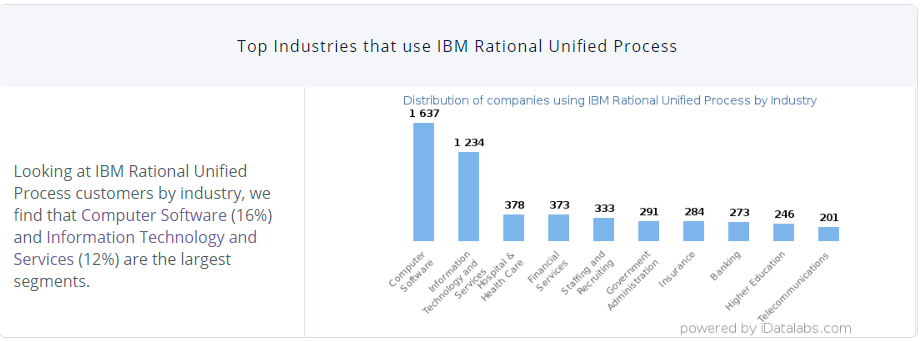
1. **Điều kiện áp dụng, các công cụ, biểu mẫu áp dụng của RUP**
   1. **Điều kiện áp dụng**

Quy trình RUP được áp dụng trong nhiều lĩnh vực, đặc biệt được áp dụng nhiều tại Mĩ

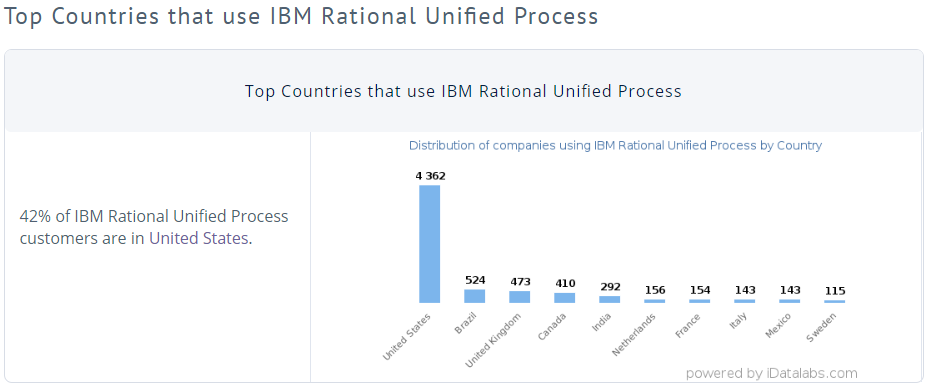
* Tỉ lệ các công ty sử dụng RUP



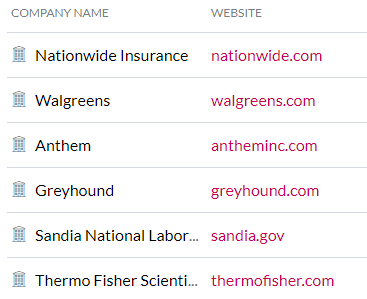
Số lượng công ty sử dụng phần mềm RUP

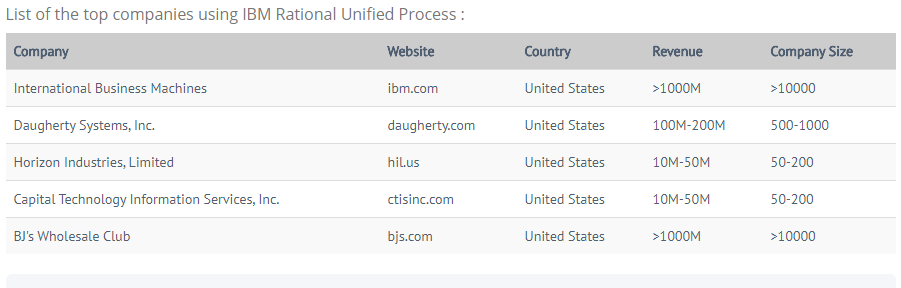


Trong đó tại Mĩ



* Dưới đây là tên một số công ty áp dụng RUP



Và 

* 1. **Các công cụ của RUP**

Để áp dụng được quy trình phát triển hệ thống phần mềm của Rational thì yêu cầu không thể thiếu là hệ thống các phần mềm công cụ hỗ trợ. Hãng Rational đã xây dựng một hệ thống công cụ như vậy, mà tiêu biểu và thường dùng nhất là:

* Phần mềm Rational Requisite Pro: cho phép phân tích các yêu cầu, xây dựng kế hoạch thực hiện, xác định các tác nhân của hệ thống cùng những tình huống sử dụng.
* Phần mềm Rational Rose: cho phép xây dựng các mô hình phân tích, thiết kế, triển khai.
* Phần mềm Rational XDE: cho phép vừa xây dựng các mô hình vừa kết sinh mã nguồn chương trình song song với nhau
* Phần mềm Rational Clear Case: quản trị dự án phân tích thiết kế, cho phép làm việc theo nhóm.
  1. **Biểu mẫu áp dụng**

RUP áp dụng cho nhiều công việc vì thế cũng có nhiều biểu mẫu khác nhau. Dưới đây là một số biểu mẫu riêng cho từng ngành học

* Business Related Templates

|  |  |
| --- | --- |
| File | **Template Name** |
| [rup\_bvis.dot](http://jdbv.sourceforge.net/templates/rup_bvis.dot) | Business Vision |
| [rup\_barchdoc.dot](http://jdbv.sourceforge.net/templates/rup_barchdoc.dot) | Business Architecture Document |
| [rup\_bgloss.dot](http://jdbv.sourceforge.net/templates/rup_bgloss.dot) | Business Glossary |
| [rup\_blomtl.dot](http://jdbv.sourceforge.net/templates/rup_blomtl.dot) | Bill of Materials |
| [rup\_bmgd.dot](http://jdbv.sourceforge.net/templates/rup_bmgd.dot) | Business Modeling Guidelines |
| [rup\_brul.dot](http://jdbv.sourceforge.net/templates/rup_bmgd.dot) | Business Rules |
| [rup\_bucs.dot](http://jdbv.sourceforge.net/templates/rup_bucs.dot) | Business Use-Case Specification |
| [rup\_sbs.dot](http://jdbv.sourceforge.net/templates/rup_sbs.dot) | Supplementary Business Specification |
| [up\_bucr.dot](http://jdbv.sourceforge.net/templates/rup_bucr.dot) | Business Use-Case Realization Specification |
| [rup\_buscs.dot](http://jdbv.sourceforge.net/templates/rup_buscs.dot) | Business Case |
| [rup\_tarorgass.dot](http://jdbv.sourceforge.net/templates/rup_tarorgass.dot) | Target-Organization Assessment |

* Project Management Templates

|  |  |
| --- | --- |
| File | **Template Name** |
| [rup\_vision.dot](http://jdbv.sourceforge.net/templates/rup_vision.dot) | Vision |
| [rup\_vision\_sp.dot](http://jdbv.sourceforge.net/templates/rup_vision_sp.dot) | Vision (Small Project) |
| [rup\_sdpln.dot](http://jdbv.sourceforge.net/templates/rup_sdpln.dot) | Software Development Plan |
| [rup\_sdpln\_sp.dot](http://jdbv.sourceforge.net/templates/rup_sdpln_sp.dot) | Software Development Plan (Small Project) |
| [rup\_gloss.dot](http://jdbv.sourceforge.net/templates/rup_gloss.dot) | Glossary |
| [rup\_devcs.dot](http://jdbv.sourceforge.net/templates/rup_devcs.dot) | Development Case |
| [rup\_dorgass.dot](http://jdbv.sourceforge.net/templates/rup_dorgass.dot) | Development-Organization Assessment |
| [rup\_rsklst.dot](http://jdbv.sourceforge.net/templates/rup_rsklst.dot) | Risk List |
| [rup\_rskpln.dot](http://jdbv.sourceforge.net/templates/rup_rskpln.dot) | Risk Management Plan |
| [rup\_itpln.dot](http://jdbv.sourceforge.net/templates/rup_itpln.dot) | Iteration Plan |
| [rup\_itass.dot](http://jdbv.sourceforge.net/templates/rup_itass.dot) | Iteration Assessment |
| [rup\_mspln.dot](http://jdbv.sourceforge.net/templates/rup_mspln.dot) | Measurement Plan |
| [rup\_stass.dot](http://jdbv.sourceforge.net/templates/rup_stass.dot) | Status Assessment |
| [rup\_prspln.dot](http://jdbv.sourceforge.net/templates/rup_prspln.dot) | Problem Resolution Plan |

* Requirements

|  |  |
| --- | --- |
| File | **Template Name** |
| [rup\_rmpln.dot](http://jdbv.sourceforge.net/templates/rup_rmpln.dot) | Requirements Management Plan |
| [rup\_srs.dot](http://jdbv.sourceforge.net/templates/rup_srs.dot) | Software Requirements Specification for Subsystem |
| [rup\_srsuc.dot](http://jdbv.sourceforge.net/templates/rup_srsuc.dot) | Software Requirements Specification For Subsystem or Feature |
| [rup\_ucmgd.dot](http://jdbv.sourceforge.net/templates/rup_ucmgd.dot) | Use-Case-Modeling Guidelines |
| [rup\_ucspec.dot](http://jdbv.sourceforge.net/templates/rup_ucspec.dot) | Use-Case Specification |
| [rup\_ucrs.dot](http://jdbv.sourceforge.net/templates/rup_ucrs.dot) | Use-Case-Realization Specification |
| [rup\_sspec.dot](http://jdbv.sourceforge.net/templates/rup_sspec.dot) | Supplementary Specification |

* Development

|  |  |
| --- | --- |
| File | **Template Name** |
| [rup\_desgd.dot](http://jdbv.sourceforge.net/templates/rup_desgd.dot) | Design Guidelines |
| [rup\_prggd.dot](http://jdbv.sourceforge.net/templates/rup_prggd.dot) | Programming Guidelines |
| [rup\_sad.dot](http://jdbv.sourceforge.net/templates/rup_sad.dot) | Software Architecture Document |
| [rup\_ibpln.dot](http://jdbv.sourceforge.net/templates/rup_ibpln.dot) | Integration Build Plan |

* Testing and Quality

|  |  |
| --- | --- |
| File | **Template Name** |
| [rup\_tsteval.dot](http://jdbv.sourceforge.net/templates/rup_tsteval.dot) | Test Evaluation Summary |
| [rup\_tstgd.dot](http://jdbv.sourceforge.net/templates/rup_tstgd.dot) | Test Guidelines |
| [rup\_tstpln.dot](http://jdbv.sourceforge.net/templates/rup_tstpln.dot) | Iteration or Master Test Plan |
| [rup\_qapln.dot](http://jdbv.sourceforge.net/templates/rup_qapln.dot) | Quality Assurance Plan |
| [TestCase.dot](http://jdbv.sourceforge.net/templates/TestCase.dot) | Test Case |

* Deployment

|  |  |
| --- | --- |
| File | **Template Name** |
| [rup\_pacpln.dot](http://jdbv.sourceforge.net/templates/rup_pacpln.dot) | Product Acceptance Plan |
| [rup\_cmpln.dot](http://jdbv.sourceforge.net/templates/rup_cmpln.dot) | Configuration Management Plan |
| [rup\_dplpln.dot](http://jdbv.sourceforge.net/templates/rup_dplpln.dot) | Deployment Plan |
| [rup\_relnt.dot](http://jdbv.sourceforge.net/templates/rup_relnt.dot) | Release Notes |

1. **Lợi ích và hạn chế của RUP**
   1. **Lợi ích**

* Phát triển phần mềm theo vòng lặp " Các phần được lên kế hoạch dựa vào độ ưu tiên của khách hàng và phân phối những phần có độ ưu tiên cao nhất trước”.
* Quản lý yêu cầu " Viết tài liệu một cách rõ ràng cho các yêu cầu khách hàng và theo dõi sự thay đổi của những yêu cầu này”.
* Sử dụng kiến trúc dựa vào component " Tổ chức hệ thống như một tập các component có thể tái sử dụng”.
* Mô hình hóa phần mềm một cách trực quan " Sử dụng các mô hình đồ họa UML để biểu diễn các góc nhìn tĩnh và động của phần mềm”.
* Kiểm tra chất lượng phần mềm " Đảm bảo rằng phần mềm đáp ứng được các chuẩn chất lượng về mặt tổ chức”.
* Điều khiển các thay đổi phần mềm " Quản lý những thay đổi phần mềm sử dụng những hệ thống quản lý thay đổi và các công cụ quản lý cấu hình”.
  1. **Hạn chế**
* Tiến trình nặng.
* Các tiến trình dự án rất phức tạp để thực hiện vì phải xây dựng quá trình thiết kế cụ thể.
* Để sử dụng thì người tham gia dự án phải học cách làm việc.
* Quá trình phát triển có thể vượt quá tầm kiểm soát (do đánh giá ban đầu sai về chi phí, tài nguyên và rủi ro cũng như do các yếu tố bất định).
* Cần chuyên gia để có thể đáp ứng được các mục tiêu của mô hình phát triển này.

1. **Tài liệu tham khảo**

<https://vi.wikipedia.org/wiki/Quy_tr%C3%ACnh_ph%C3%A1t_tri%E1%BB%83n_ph%E1%BA%A7n_m%E1%BB%81m>

<https://ndkeis.blogspot.com/2011/09/quy-trinh-rup-rational-unified-process.html>

<https://www.ibm.com/support/knowledgecenter/en/SSSHCT_7.1.0/com.ibm.reqpro.help/administering/projects/creating_modifying/r_rup_proj_template.html>

<https://toc.123doc.org/document/363032-5-rup-rational-unified-process-sdlc.htm>

<http://files.defcon.no/RUP/process/templates.htm>

<https://text.123doc.org/document/2237761-tim-hieu-ve-quy-trinh-phat-trien-phan-mem-rup.htm>

https://www.slideshare.net/loixua87u/bai-giangjava2008120920104133phpapp01