

**Pró-Reitoria Acadêmica  
Escola de Educação, Tecnologia e Comunicação  
Curso de Bacharelado em Engenharia de Software**

**Trabalho da Disciplina de Engenharia de Software**

**FASES DO CICLO DE VIDA DO (*RATIONAL UNIFIED PROCESS – IRUP*)**

**Autores:** Natanael Ferreira Neves  
Wagner Serpa Porto

**Orientador:** Prof. Dr. Milton Pombo da Paz

**Natanael Ferreira Neves**

**Wagner Serpa Porto**

**FASES DO CICLO DE VIDA DO (*RATIONAL UNIFIED PROCESS – IRUP*)**

Documento apresentado ao Curso de graduação de Bacharelado em Engenharia de Software da Universidade Católica de Brasília, como requisito parcial para obtenção da aprovação na disciplina de Engenharia de Software.

Orientador: Prof. Dr. Milton Pombo da Paz

**Brasília  
2024**

## RESUMO

O *Rational Unified Process RUP* se trata de uma metodologia estruturada para o desenvolvimento de software, na qual possui orientações claras para cada fase do ciclo de vida do projeto, onde que a sua variante, o *Incremental Rational Unified Process iRUP*, adapta o *RUP* para um modelo que de certa forma é mais ágil e incremental, permitindo maior flexibilidade durante o desenvolvimento, onde que os mesmos utilizam técnicas de prevenção de riscos como *brainstorming* e análise *SWOT*, sendo assim os riscos são identificados e mitigados, garantindo maior assertividade no planejamento, promovendo a colaboração e a qualidade, assegurando que o produto final atenda às expectativas do cliente.

**Palavras-chave:** Metodologia, Ciclo de vida, Variante, Ágil.

## ***ABSTRACT***

*The Rational Unified Process RUP is a structured methodology for software development, which has clear guidelines for each phase of the project life cycle, where its variant, the Incremental Rational Unified Process iRUP, adapts the RUP to a model that is in a way more agile and incremental, allowing greater flexibility during development, where they use risk prevention techniques such as brainstorming and SWOT analysis, thus risks are identified and mitigated, ensuring greater assertiveness in planning, promoting collaboration and quality, ensuring that the final product meets customer expectations.*

**Keywords:** *Methodology, Lifecycle, Variant, Agile.*

## SUMÁRIO

|  |           |
|--|-----------|
| <b>RESUMO .....</b>  | <b>3</b>  |
| <b>ABSTRACT .....</b>  | <b>4</b>  |
| <b>1 INTRODUÇÃO.....</b>   | <b>6</b>  |
| <b>2 CICLO DE VIDA DO <i>IRUP</i> .....</b>  | <b>7</b>  |
| 2.1 FASE DE CONCEPÇÃO DO <i>INCREMENTAL RATIONAL UNIFIED PROCESS “IRUP”</i> .....  | 7         |
| 2.2 DEFINIÇÃO DO ESCOPO DO PROJETO .....   | 8         |
| 2.3 IDENTIFICAÇÃO DOS <i>STAKEHOLDERS</i> .....                                    | 8         |
| 2.4 DEFINIÇÃO DE REQUISITOS INICIAIS .....   | 8         |
| 2.5 PLANEJAMENTO PRELIMINAR .....  | 9         |
| 2.6 IDENTIFICAÇÃO DE RISCOS INICIAIS .....   | 9         |
| 2.7 FASE DE ELABORAÇÃO DO <i>INCREMENTAL RATIONAL UNIFIED PROCESS “IRUP”</i> ..... | 10        |
| 2.8 ANÁLISE E DETALHAMENTO DOS REQUISITOS .....                                    | 10        |
| 2.9 DESENVOLVIMENTO DA ARQUITETURA DO SISTEMA .....                                | 10        |
| 2.10 IDENTIFICAÇÃO E MITIGAÇÃO DE RISCOS .....                                     | 11        |
| 2.11 PLANEJAMENTO DETALHADO .....  | 11        |
| <b>3 CONCLUSÃO.....</b>  | <b>12</b> |
| <b>REFERÊNCIAS .....</b>   | <b>13</b> |

## 1 INTRODUÇÃO

Neste tópico introdutório será apresentada uma visão geral sobre o *RUP* ou *Rational Unified Process*, onde que traduzido para língua portuguesa significa processo unificado, no qual se trata de uma metodologia ou *framework* que foi desenvolvida na empresa *Rational*, com o intuito de apoiar o desenvolvimento de software orientado ao objeto na qual proporciona vantagens na utilização da *UML* “*Unified Modeling Language*” (VETORAZZO, 2018).

De acordo com Kruchten (2000) e Pressman, Maxim (2021), deve se observar que embora o conceito de *iRUP Incremental Rational Unified Process*, não seja amplamente documentado como uma metodologia oficial separada, o mesmo é visto como uma variante mais ágil e adaptativa do RUP, e as práticas associadas ao RUP podem ser ajustadas para torná-las mais incrementais e ágeis em comparação ao *RUP*, mas ambas possui a mesma base teórica.

Como já foi mencionado anteriormente o *RUP* se trata de uma metodologia ou *framework*, onde que as metodologias dentro do campo da engenharia de software servem como alicerces para um processo de engenharia de software completo através da identificação da quantidade de atividades metodológicas que são possivelmente aplicáveis a projetos de software, independentemente da complexidade e do tamanho, sendo assim, a metodologia de processo engloba um conjunto de atividades de apoio que são aplicáveis a todo o processo de software onde que toda metodologia de processo tem em comum cinco atividades, sendo elas comunicação, planejamento, modelagem e entrega (PRESSMAN; MAXIM, 2021).

Os processos de software são importantes, pois proporciona para os membros da equipe de projeto um direcionamento sobre o que deve ser realizado para atender aos objetivos do software, ou seja, os processos de software definem quais são as atividades que permitem obter o produto de software, onde que observa se que o processo Cascata ou *Waterfall Model*, proposto na década de 1970, reflete sobre o que é mais utilizado em outros projetos de engenharia, sendo aplicado em desenvolvimento de software, onde que o processo unificado *RUP* “*Rational Unified Process*”, “*IBM/Rational*”, abrange as fases de concepção, elaboração, construção e transição (HIRAMA, 2011).

O *RUP* e derivado do *UP* ou *Unified Process*, no qual o mesmo destaca as fases que correspondem à parte dinâmica do processo e os fluxos de trabalho que correspondem à parte dinâmica do processo, em que diferentemente dos processos em cascata, evolutivo e modelo V, o *UP* reúne as fases e os fluxos de trabalho em um único processo, sendo assim o *UP* foi proposto no cenário da abordagem Orientada a Objetos, onde que o processo é dirigido a casos

de uso, no qual em seguida, uma estratégia de projeto é definida a partir dos Casos de Uso mais críticos, sendo assim o *RUP* derivado dos trabalhos *UP* e *UML* (PRESSMAN, 2007).

Uma das vantagens do que pode se destacar do *RUP* em relação aos demais processos de engenharia de software é a desvinculação das fases com os fluxos de trabalho que podem requisitados em todos os estágios do processo, sendo que o reconhecimento da implantação de software no ambiente do usuário fazendo parte do processo, mas observa-se que o *RUP* não é adequado para todos os tipos de desenvolvimento de software, tendo limitações para lidar com softwares que possuem a natureza de processarem uma grande quantidade de dados, como na área de transportes, comunicações etc, porque o mesmo trata o software um produto de engenharia na qual o mesmo detalha e amplia o material técnico, complementando-o com aspectos gerenciais, sendo assim o *RUP* não deve ser confundido com o processo unificado *PU*, pois o mesmo possui estrutura diferente, utilizando as mesmas fases e as mesmas características básicas, mas as atividades técnicas são divergentes, sendo denominadas na mesma como disciplinas, tornando o *RUP* uma coleção de processos completos (HIRAMA, 2011).

## 2 CICLO DE VIDA DO *iRUP*

No campo da engenharia de software, os processos podem ser definidos por diversas atividades como manutenção, contratação, desenvolvimento e aquisição de software, onde se pode definir também subprocessos para cada uma dessas atividades, sendo assim em um processo de desenvolvimento de software, o ponto de partida para a concepção de um processo é a escolha de um modelo de ciclo de vida (FILHO, 2019).

E como foi mencionado anteriormente de acordo com Hirama (2011), o ciclo de vida do *RUP* possui quatro fases, sendo elas denominadas como, concepção, elaboração, construção e transição, onde que o *iRUP* por ser uma variante do *RUP*, faz com que o mesmo utilize a mesma base teórica.

### 2.1 FASE DE CONCEPÇÃO DO *INCREMENTAL RATIONAL UNIFIED PROCESS* “*iRUP*”

A fase de iniciação ou concepção, é a primeira etapa do Rational Unified Process (*RUP*) e tem como objetivo principal obter um diálogo com o cliente de modo a entender suas necessidades e expectativas em relação ao software, estabelecer o propósito do projeto e obter uma análise preliminar a fim de avaliar sua viabilidade (IBM, 2021).

Para esta fase, existem tarefas como a definição do escopo do projeto, identificação dos stakeholders, definição de requisitos iniciais, planejamento preliminar, identificação de riscos iniciais, artefatos gerados (IBM, 2021), onde os mesmos serão brevemente descritos a seguir no próximos itens.

## 2.2 DEFINIÇÃO DO ESCOPO DO PROJETO

A definição do escopo do projeto se inicia a partir do estabelecimento dos limites do sistema a ser desenvolvido, incluindo as funcionalidades essenciais e o que ficará fora do escopo, onde que essa definição ajuda a evitar retrabalho futuro, facilita o planejamento de recursos e permite priorizar as tarefas de desenvolvimento (SOMMERVILLE, 2007).

Sendo assim exemplificadamente em um projeto de automação de segurança, o escopo pode incluir funcionalidades como reconhecimento facial e abertura automática de portões eletrônicos, enquanto o uso de inteligência artificial para identificar comportamentos ofensivos e realizar tratativas junto à polícia podem ser implementadas em *features* futuros (SOMMERVILLE, 2007).

## 2.3 IDENTIFICAÇÃO DOS *STAKEHOLDERS*

A identificação dos *stakeholders* é focada na identificação de todos os *stakeholders* envolvidos, desde o usuário do sistema, clientes, *product owner* e equipe técnica, com o objetivo de garantir que o sistema atenda às expectativas de todos os envolvidos e que os requisitos sejam definidos de forma precisa, onde que uma das tarefas para esta definição é conduzir entrevistas ou *workshops* com *stakeholders* para compreender suas dúvidas e questionamentos (VETORAZZO, 2018).

## 2.4 DEFINIÇÃO DE REQUISITOS INICIAIS

Após identificado as demandas, dúvidas e questionamentos do cliente, a definição de requisitos visa o levantamento dos requisitos funcionais e não funcionais essenciais para se atingir o objetivo do projeto, com foco nos casos de uso mais importantes (IBM, 2021).

Prototipa-se uma base inicial, permitindo que a equipe técnica compreenda as necessidades do negócio e identifique riscos e refine os conceitos e defina propriedades, onde que a partir desta tarefa, é gerado um documento, que lista os requisitos principais e como o sistema atenderá a essas necessidades (PRESSMAN, 2007).



## 2.5 PLANEJAMENTO PRELIMINAR

O planejamento preliminar é a elaboração de um plano inicial para o projeto, incluindo estimativas de custo, estipulação de prazos e alocação de recursos, onde o mesmo visa dar liberalidade aos envolvidos contemplarem de forma mais palpável como será a evolução do projeto, seus custos, cronograma, objetivando disponibilizar clareza e facilitar aprovação do projeto, sendo assim um processo iterativo e incremental, o artefato aqui gerado pode ser dividido em ciclos, detalhando metas para cada ciclo (KRUCHTEN, 2000).

## 2.6 IDENTIFICAÇÃO DE RISCOS INICIAIS

A identificação dos riscos iniciais se trata da identificação e documentação dos possíveis riscos ao projeto, sejam técnicos, financeiros ou relacionados a prazos, sendo assim o intuito é facilitar a criação de estratégias para mitigar a probabilidade de falhas durante o desenvolvimento (IBM, 2021).

Sendo assim a falta de clareza na comunicação com o cliente, tornará o desenvolvimento oneroso tanto para o cliente quanto para a equipe de desenvolvimento visto que, ao desenvolver um software sem a devida documentação, ocorre o grande risco de perda de prazos e recursos causados pelo risco do retrabalho onde o mesmo deve ser evitado ao máximo, onde que para isso é utilizado técnicas de identificação de riscos como Brainstorming, na qual é uma técnica que é realizada uma reunião entre os membros da equipe para gerar ideias sobre possíveis riscos, na qual não pode ser realizada sem julgamento, onde que outra técnica utilizada é o *SWOT Strengths, Weaknesses, Opportunities, Threats*, onde a mesma é uma técnica que ajuda a identificar riscos em relação a forças, fraquezas, oportunidades e ameaças no contexto do projeto, e outra técnica bastante utilizada é a revisão de projetos passados, na qual consiste em estudar projetos similares ou projetos anteriores para identificar riscos comuns que ocorreram em situações semelhantes. (VOITTO, 2020).

Sendo assim o enfoque desta etapa é objetivado em adquirir os *insights* necessários para um desenvolvimento escalonável, assertivo e eficaz, seja por meio de entrevistas individuais ou em grupo, a clareza das necessidades do cliente deve ser aqui esclarecida, onde que a fase de concepção é concluída com a aprovação do caso de negócio e do escopo inicial pelos *stakeholders*, onde que esse marco é denominado de *Lifecycle Objective Milestone* no *RUP*, e garante que o projeto tenha viabilidade técnica e de negócio antes de seguir para a fase de Elaboração (PRESSMAN, MAXIN 2007; HIRAMA 2011).

## 2.7 FASE DE ELABORAÇÃO DO *INCREMENTAL RATIONAL UNIFIED PROCESS* “iRUP”

A fase de elaboração tem por objetivo analisar o domínio do problema, de modo a propor uma arquitetura para os cenários da aplicação, sendo assim o projeto é planejado e especificado de acordo com suas características, havendo uma concentração maior nos detalhes da análise do problema de forma que um plano de projeto seja desenvolvido, sendo assim são eliminados os elementos de maior risco para o bom desenvolvimento do projeto, pois esta fase busca analisar o domínio do problema, para se estabelecer uma fundamentação arquitetônica que sirva para desenvolver o plano de projeto, sendo assim a mesma possui quatro atividades principais que são a, análise e detalhamento dos requisitos, desenvolvimento da arquitetura, identificação e mitigação dos riscos e por fim o planejamento detalhado (MARINHO, 2003).

## 2.8 ANÁLISE E DETALHAMENTO DOS REQUISITOS

Na atividade de análise e detalhamento dos requisitos é realizada um refino dos requisitos iniciais que foram levantados na fase de Iniciação, ou seja, ocorre um detalhamento que busca esclarecer todas as dúvidas, onde que para isso se faz do uso de técnicas como entrevistas com usuários, *workshops*, protótipos e análise de casos de uso, com o objetivo de transformar os requisitos de alto nível em especificações que sejam mais claras e precisas, de forma que a equipe de desenvolvimento tenha uma visão clara do que precisa ser feito realizado, sendo assim a mesma possui o objetivo de garantir que o sistema seja desenvolvido de acordo com as expectativas e necessidades dos *stakeholders* (JACOBSON et al, 1999).

## 2.9 DESENVOLVIMENTO DA ARQUITETURA DO SISTEMA

A atividade de desenvolvimento da arquitetura do sistema dentro da fase de elaboração tem o objetivo de garantir que o sistema será capaz de atender aos requisitos que foram pré-definidos, com qualidade e eficiência, sendo assim nesta atividade, são realizadas as primeiras decisões sobre as tecnologias, plataformas, *frameworks* e design de alto nível, incluindo a divisão do sistema em componentes ou módulos, onde que para que isso ocorra a arquitetura proposta precisa ser escalável, permitindo futuras alterações sem comprometer a estabilidade do sistema, onde que durante essa atividade é criado protótipos e diagramas de arquitetura, como diagramas de componentes e de classes, para validar as escolhas feitas anteriormente, sendo realizados testes iniciais para validar se a arquitetura atende os requisitos não funcionais, como desempenho e segurança, e se o sistema poderá ser mantido e expandido ao longo do tempo (KRUCHTEN, 2000).

## 2.10 IDENTIFICAÇÃO E MITIGAÇÃO DE RISCOS

A atividade de identificação e mitigação de riscos dentro da fase de elaboração é uma atividade que possui o objetivo de prever possíveis problemas que possam surgir durante o desenvolvimento, ou seja a mesma tem a finalidade de buscar soluções ou estratégias para minimizá-los o máximo possível (PRESSMAN, 2007).

Os riscos no desenvolvimento de software podem ser de diversas naturezas, como riscos tecnológicos, requisitos, financeiros, cronograma e entre outros, sendo assim durante essa atividade, a equipe realiza uma análise detalhada dos pontos que são considerados mais críticos do projeto, como a escolha de tecnologias, a complexidade dos requisitos ou o impacto de decisões no orçamento e no cronograma, onde que após identificado os riscos, a equipe responsável desenvolve planos para combater para cada um deles, podendo ser ações corretivas, ajustes no planejamento ou até mesmo mudanças na abordagem do projeto, onde que deve se observar que a mitigação de riscos é uma atividade contínua e iterativa, que se estende até as fases seguintes do desenvolvimento (AMBLER, 2002).

## 2.11 PLANEJAMENTO DETALHADO

A atividade do planejamento detalhado na fase de Elaboração é essencial para garantir que o projeto tenha uma execução organizada e eficiente nas fases que se seguem, na qual durante essa atividade a equipe responsável pelo gerenciamento revisa o cronograma inicial levando em conta as informações obtidas nas fases anteriores, como os requisitos que foram refinados anteriormente e os riscos mitigados (PRESSMAN, 2007).

Essa atividade possui o objetivo de criar um plano de ação detalhado o suficiente para a fase de Construção, na qual o mesmo contenha uma definição clara das atividades, marcos e demais elementos que devem ser entregues de forma que os recursos estejam alocados de maneira correta, sendo assim o planejamento envolve a definição de cronogramas de entrega e as dependências entre as atividades que devem ser desenvolvidas e suas respectivas distribuições de responsabilidades entre os membros da equipe, onde que também são identificados os recursos necessários para o desenvolvimento das mesmas, como ferramentas, tecnologias, especialistas e orçamento, sendo assim com esse planejamento refinado, o mesmo ajuda a equipe a se preparar para os desafios futuros do desenvolvimento e a manter o projeto dentro dos limites de tempo e custo (JACOBSON et al, 1999).

### 3 CONCLUSÃO

Conclui-se que o *Rational Unified Process RUP* e sua variante *Incremental Rational Unified Process iRUP* são abordagens sólidas para o desenvolvimento de software, oferecendo uma estrutura sistemática e organizada que garante a entrega de sistemas software de alta qualidade, sendo assim se destacando pela suas capacidades de organizar e guiar o processo de desenvolvimento de software por meio de suas fases e fluxos de trabalho, adaptando-se a diferentes necessidades de projeto.

## REFERÊNCIAS

AMBLER, Scott W. Agile Modeling: Effective Practices for extreme Programming and the Unified Process. John Wiley & Sons, 2002.

Aula 03 - RUP — Nickerson Ferreira. Disponível em:

<<https://docentes.ifrn.edu.br/nickersonferreira/disciplinas/projeto-de-desenvolvimento-de-software/aula-07-fase-de-projeto/view>>. Acesso em: 18 nov. 2024.

Aula 07 - Fase de Projeto — Nickerson Ferreira. Disponível em:

<<https://docentes.ifrn.edu.br/nickersonferreira/disciplinas/projeto-de-desenvolvimento-de-software/aula-07-fase-de-projeto/view>>. Acesso em: 18 nov. 2024.

FILHO, Wilson de Pádua P. Engenharia de Software - Produtos - Vol.1. 4th ed. Rio de Janeiro: LTC, 2019. E-book. p.61. ISBN 9788521636724. Disponível em:

<https://integrada.minhabiblioteca.com.br/reader/books/9788521636724/>. Acesso em: 19 nov. 2024.

HIRAMA, Kechi. Engenharia de Software. Rio de Janeiro: GEN LTC, 2011. E-book. p.62. ISBN 9788595155404. Disponível em:

<https://integrada.minhabiblioteca.com.br/reader/books/9788595155404/>. Acesso em: 17 nov. 2024.

IBM, Engineering Method Composer. Disponível em: <<https://www.ibm.com/docs/pt-br/engineering-lifecycle-management-suite/lifecycle-optimization-method-composer/7.6.0?topic=composer-overview-engineering-method>>.

Acesso em: 19 nov. 2024.

Introdução ao RUP. Disponível em: <[https://www.cin.ufpe.br/~gta/rup-vc/core.base\\_rup/guidances/supportingmaterials/introduction\\_to\\_rup\\_36B63436.html](https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/guidances/supportingmaterials/introduction_to_rup_36B63436.html)>.

JACOBSON, Ivar. The unified software development process. Pearson Education India, 1999.

JOHNSON, Thienne de Melo e Silva M.; COUTINHO, Mauro M. Avaliação de Desempenho de Sistemas Computacionais. Rio de Janeiro: LTC, 2011. E-book. p.1. ISBN 978-85-216-

2022-8. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/978-85-216-2022-8/>. Acesso em: 17 nov. 2024.

KRUCHTEN, Philippe. The rational unified process: an introduction. Addison-Wesley Professional, 2004.

MARINHO, Fabiana et al. Uma Proposta de um Repositório de Padrões de Software Integrado ao RUP. In: Anais da Terceira Conferência Latino-americana em Linguagens de Padrões para Programação (SugarLoafPLOP 2003). Porto de Galinhas, PE, Brasil. 2003.

PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de software. 9th ed. Porto Alegre: AMGH, 2021. E-book. p.1. ISBN 9786558040118. Disponível em:

<https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/>. Acesso em: 17 nov. 2024.

The Rational Objectory Process - A UML-based Software Engineering Process. Disponível em:

<[https://web.archive.org/web/20190527224552/http://www.iscn.at/select\\_newspaper/object/rational.html](https://web.archive.org/web/20190527224552/http://www.iscn.at/select_newspaper/object/rational.html)>. Acesso em: 17 nov. 2024.

VETORAZZO, Adriana S. Engenharia de software. Porto Alegre: SAGAH, 2018. E-book. p.101. ISBN 9788595026780. Disponível em:

<https://integrada.minhabiblioteca.com.br/reader/books/9788595026780/>. Acesso em: 18 nov. 2024.