41951 Análise de Sistemas - 2024/25 [ASis25]

Lab 08 - gestão ágil de projetos

Mar 31, 2025 | Este laboratório foca o papel das histórias de utilizadores (*user stories*) na gestão ágil de projetos e boas práticas para as escrever e acompanhar.

## Percurso de aprendizagem

Em grupo.

### O que é uma *user story*?

Atualmente, a generalidade dos projetos de desenvolvimento de software são geridos com práticas ditas “ágeis”, que procuram favorecer uma cultura de comunicação e flexibilidade, corresponsabilizando os membros da equipa na gestão e acompanhamento do projeto.

A organização das atividades com o framework Scrum, ou a visualização de trabalho em quadros Kanban, são exemplos de técnicas de gestão ágil de projetos, populares nos projetos de desenvolvimento de software.

Em ambos os casos, a “unidade de planeamento” (e de distribuição de trabalho) é a *user story*.

* Procure, [na documentação do produto JIRA](https://www.atlassian.com/agile/project-management), informação sobre os conceitos de *Epic* e *User* Story. Defina estes termos por palavras suas.
* Exemplifique esses conceitos, considerando, para isso, a área de aplicação de um portal para encomenda de comida. Pode estudar [este exemplo](https://docs.google.com/document/d/1To-94vkTmpyGP0hyqimxvuTOXeT86hB9Ob0DpnANvYQ/edit?tab=t.0), na área das ofertas de emprego.

### Uma *checklist* para INVESTir

Para a gestão dos projetos ser eficaz, é importante que a granularidade atribuída às stories seja comparável e adequada. Na gestão ágil, o acrónimo INVEST (ver tabela) traduz uma [*checklist* com propriedades que uma *user story* deve apresentar.](https://scrum-master.org/en/creating-the-perfect-user-story-with-invest-criteria)

| **Critério** | **Significado** | **Exemplo/notas** |
| --- | --- | --- |
| **I**ndependente | A história pode ser desenvolvida e testada de forma independente | “Como utilizador, posso redefinir a minha palavra-passe sem afetar outras funcionalidades.” |
| **N**egociável | A história capta a intenção, não um contrato fixo. Devem permanecer flexíveis para acomodar o refinamento colaborativo. | Evite especificações demasiado pormenorizadas que impeçam a colaboração. |
| Útil (***V****aluable*) | A história fornece valor ao utilizador final (contribui com um incremento útil). | “Como cliente, quero ver ofertas personalizadas.” |
| Pode-se estimar (***E****stimable*) | Uma história bem definida permite à equipa avaliar o esforço necessário para a implementação com uma precisão razoável. | A história não pode ser grande ou vaga que não possa ser dimensionada em pontos. |
| Pequena (***S****mall*) | A história é dimensionada para caber num sprint e não requer mais que até alguns dias. | O “processo de check-out” pode ser dividido em partes mais pequenas. Uma listagem com filtros pode ser dividida por tipos de filtros, com ou sem paginação,... |
| **T**estável | Uma história deve ser verificável através de critérios claros que permitam confirmar que cumpre os requisitos pretendidos | Os critérios de aceitação ou testes confirmam se a história está concluída. |

* Tendo presente a checklist INVEST, aponte “contra-exemplos” para cada uma das boas práticas, isto é, possíveis formulações de histórias que contradizem a orientação. A tabela tem um exemplo feito para o N(egotiable).

| **Critério** | **Contra-exemplo e justificação** |
| --- | --- |
| **I**ndependente | *<completar>* |
| **N**egociável | “O candidato carrega o currículo em formato PDF, selecionando o botão verde Submit”. Problema: não está focada na intenção da interação e compromete-se com escolhas específicas da interface. |
| Útil (***V****aluable*) | *<completar>* |
| Pode-se estimar (***E****stimable*) | *<completar>* |
| Pequena (***S****mall*) | *<completar>* |
| **T**estável | *<completar>* |

### Escrever histórias… de utilização

Em grupo, escolha **um** destes casos de estudo:

1. uma aplicação de **encomenda de comida**, com Cliente (Navega por restaurantes, faz encomendas, efectua pagamentos, acompanhar as entregas), Gerente de restaurante (atualiza o menu, define preços, acompanha as encomendas recebidas, marca as encomendas como preparadas), Entregador/estafeta (recebe notificações de encomendas, actualiza o estado das encomendas durante o percurso de entrega).
2. uma aplicação de **videoconferência para trabalho colaborativo** (e.g.: Zoom, Teams, Jitsi,...), com Oganizadores (agendar reuniões ou seminários, gerir a reunião, atribuir papéis aos participantes,...), Speakers (apresenat conteúdos, controlar Q&A,...), Participantes (juntar-se à reunião, interagir no chat, colocar questões,...)

* Identifique, pelo menos, 2 épicos e 6 stories. Utilize um *template* para a descrição da user story (ver seção “User story template and examples”, no final [desta página](https://www.atlassian.com/agile/project-management/user-stories) no site da Atalassian)

### Ambiente de gestão ágil do projeto

Vamos usar uma ferramenta para a gestão do projeto considerado, adotando as práticas da gestão ágil baseada em *user stories*, iterações e quadros (visuais). Assuma que o grupo é uma equipa de desenvolvimento de um projeto de software a preparar a implementação do exemplo da Atividade anterior.

Para isso, vamos utilizar o Atlassian JIRA para criar o ambiente de gestão colaborativa do projeto.

Percurso sugerido para [criar o ambiente no JIRA](https://www.atlassian.com/software/jira/free):

1. Primeiro, é preciso criar um “site” de equipa no JIRA: uma pessoa do grupo, inscreve-se no Jira (*sign-up*) e avança para o passo “Create a site”. Sugestão: use um nome baseado na identificação do grupo, e.g., *asis909ua*.
2. Completar os passos do *wizard:*

* “What kind of work do you do?” → Software development
* “How does your team plan to use Jira?” → Work in Scrum
* É criado um projeto automaticamente no site indicado antes. O nome deste projeto pode basear-se no grupo e lab, e.g.: “g909lab08”

1. A partir da vista inicial do Jira, escolha a opção “Teams > Invite people to Jira” para adicionar os restantes colegas de grupo, que devem fazer um registo no Jira.

Utilize agora o ambiente de gestão de projeto do Jira e, pelo menos, experimente:

* Backlog: criar novas histórias de utilizadores no *backlog* geral, conforme identificadas na Atividade 3. Note o botão “Create” no topo da página.
* Sprints: configurar duração indicativa e iniciar um *sprint* (=iteração); arrastar histórias do *backlog* geral para o *backlog* do *sprint*.
* Subtarefas: a título de exemplo, divida as histórias de utilizadores em subtarefas mais pequenas (e.g.: design da user interface, atualização do modelo de dados da BD,...).

### Acompanhamento colaborativo

Continue o exercício anterior e complete a informação das users stories do projeto “lab08”, para incluir:

* [Estime uma pontuação](https://www.lucidchart.com/blog/fibonacci-scale-for-agile-estimation) para as suas *user stories*.
* Defina critérios de aceitação, com base em exemplos concretos, para estabelecer as condições em que cada história pode ser dada como completa. Adopte o [padrão GWT (Given-When-Then)](https://guides.visual-paradigm.com/give-when-then-acceptance-criteria-for-user-stories-in-agile-development/) para construir exemplos de cenários concretos.
* Experimente a atribuição de itens de trabalho (a membros da equipa).
* Arranque algumas dessas *stories* (devem estar na iteração atual) e verifique a atualização dos quadros (vista *Board*).
* Simule que já completou um par de *user stories*. Verifique a evolução do “[burn up chart](https://www.atlassian.com/agile/project-management/burn-up-chart)” (ative os relatório nos *settings* do projeto).

## Explicador

*Casos de utilização vs Stories vs Épicos*

Os **Casos de utilização** são normalmente mais formais, descrevendo na totalidade os caminhos de interação entre os utilizadores e um sistema (incluindo fluxos alternativos, condições prévias e pós-condições). Abrangem todas as maneiras como uma interação pode decorrer. São principalmente desenvovlidos em etapas de Análise.

As histórias de utilizador (***user stories****)*, pelo contrário, são mais concisas, centrando-se num único objetivo na perspetiva de um utilizador (“Como utilizador, quero...”). Definem o âmbito do que precisa de ser construído em linguagem simples, em vez de mapear todos os caminhos possíveis.

Um **épico** é um corpo de trabalho significativo que representa um tema/*milestone* funcional do produto que, normalmente, se estende por vários *sprints* e requer um esforço significativo para ser entregue ao promotor.

Tipicamente, do mais abrangente para o mais particular, teremos: épico > caso de utilização > história.

## Como aplicar os temas desde lab no projeto?

No *site* já criado no Jira (neste Lab), crie um novo projeto, baseado no template SCRUM, correspondente ao projeto do grupo (pode dar-lhe o nome baseado no produto que vai ser desenvolvido).

A partir de agora, todo o trabalho de desenvolvimento do projeto deve ser planeado no JIRA.

No registo de itens de trabalho, notar a distinção entre [“tasks” e “stories”](https://support.atlassian.com/jira-cloud-administration/docs/what-are-issue-types/#Default-work-types). As *stories* correspondem a funcionalidade observável no produto pelo utilizador final e, portanto, só se aplicam quando estamos a construir incrementos para o software. Trabalho como “preparar o modelo de casos de uso” não é uma *story*, é uma *task* (para o Jira).