

Contexte et énoncé

Vous avez exécuté les tests du projet Bibliothèque, à l'aide de la suite de tests Zephyr dans Jira.

Le projet [Évaluation Finale – Bibliothèque](#) compile les résultats de vos activités, pour les 3 derniers sprints, le sprint 4 est présentement actif.

En vue de la rétrospective de sprint prévue dans votre entreprise, vous devez justifier les résultats obtenus jusqu'à maintenant. Pour ce faire, analysez les principaux rapports dont vous disposez, afin de trouver les arguments (positifs ou négatifs) justifiant le travail de votre équipe, et faire un résumé du déroulement des activités de tests.

À ce titre, concevez un document qui servira de gabarit pour tous les résumés des activités de tests de votre entreprise. Tous les résumés doivent contenir les informations qui sont demandées dans le présent document (toutes les questions inscrites ici-bas doivent être répondues à chaque rétrospective).

Utilisez ce gabarit pour répondre aux questions qui vous sont posées.

Vous devez également ajouter une section pour l'approbation de votre résumé de sprint, par les personnes concernées.

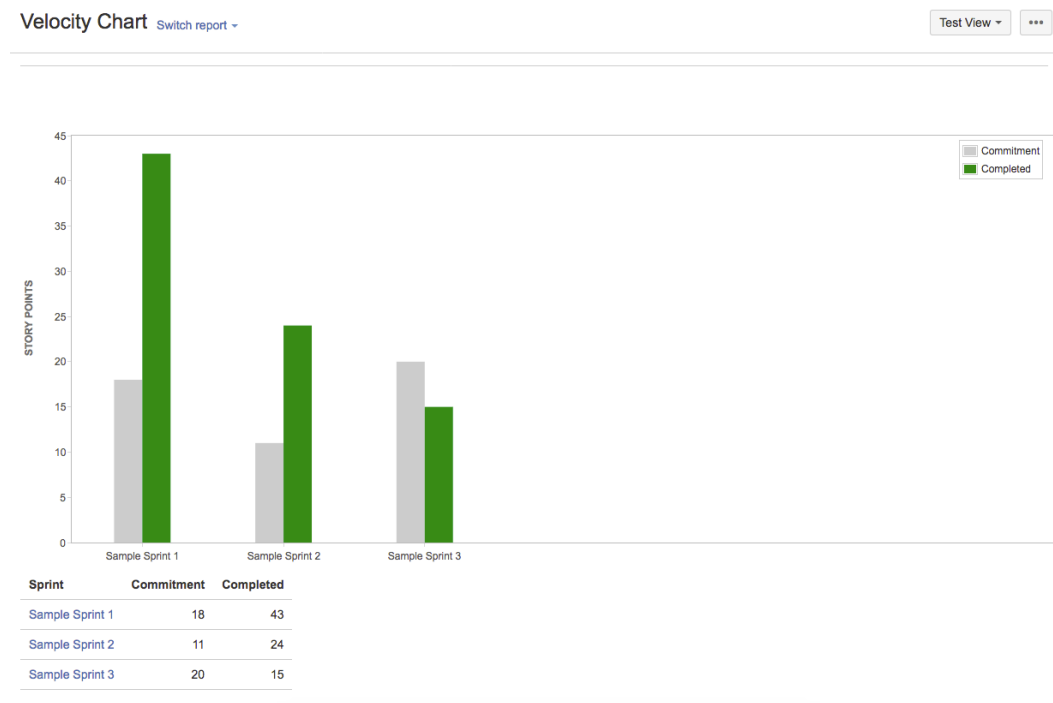
Vous serez notés sur les points suivants :

7. Collecter des mesures de la qualité
 - 7.1. Analyse complète des données à l'aide de l'outil d'emmagasiner des données
 - 7.2. Interprétation adéquate des résultats de la mesure
 - 7.3. Présentation claire des résultats de la mesure
8. Rédiger les rapports de tests
 - 8.1. Documentation exhaustive de l'avancement des activités de tests sur une base régulière (cas de tests exécutés, taux de réalisation, taux de couverture des requis etc.).
 - 8.2. Documentation exhaustive des résultats obtenus (échecs, réussites, anomalies, etc.).
 - 8.3. Analyse et évaluation claire des résultats obtenus.
 - 8.4. Identification correcte des tendances dans les anomalies.
 - 8.5. Recommandations appropriées pour la mise en production.
9. Faire approuver un rapport de tests

Données disponibles

1. Vélacité

Le [Graphique de Vélacité](#) nous montre les cas de tests effectués versus ceux planifiés :



L'équipe de développement a obtenu également un graphique similaire. Lorsque vous avez discuté avec le scrum master de l'équipe de développement, il vous a expliqué que les tâches des programmeurs avaient été mal planifiées. Ils ont donc effectué beaucoup plus de développement (tâches) que prévu les 2 premiers sprints, en conséquence des ajustements ont été apportés, mais le 3^{ième} sprint a été surestimé, et ils n'ont pas réussi à effectuer toutes les tâches prévues.

Les tests avaient été prévus en conséquence du développement (si une tâche de développement était terminée, les testeurs pouvaient alors effectuer les tests requis, et ceux-ci devaient être effectués dans le même sprint).

- Pour quelles raisons y a-t-il eu beaucoup plus de tests effectués que les prévisions initiales pour les 2 premiers sprints?
- Pour quelles raisons tous les tests n'ont-ils pas été effectués au dernier sprint?

2. Exécution de tests

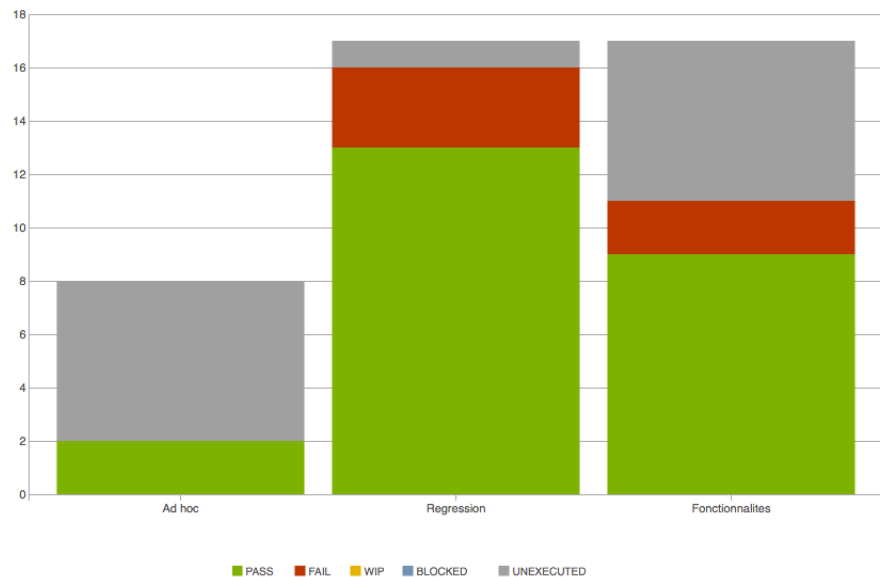
Le graphique [d'exécution des tests](#), quant à lui, nous montre une compilation différents tests exécutés, ainsi que les résultats de ceux-ci, en fonction des cycles (types) de tests, pour la version actuelle en développement (version 0.02), donc pour tous les sprints concernés par cette version (les 3 premiers dans ce cas-ci);

Test Execution Chart

[Back](#)

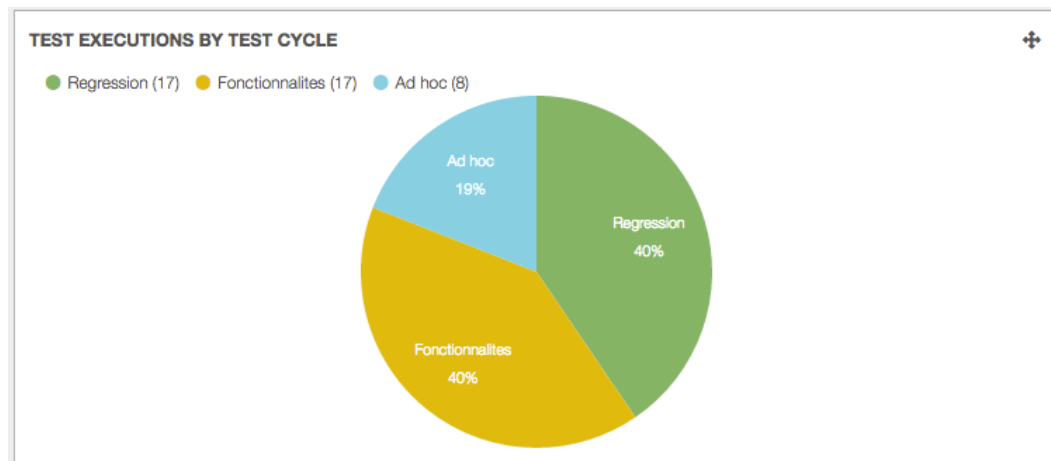
SP / Unscheduled

Test Execution Report for Cycle



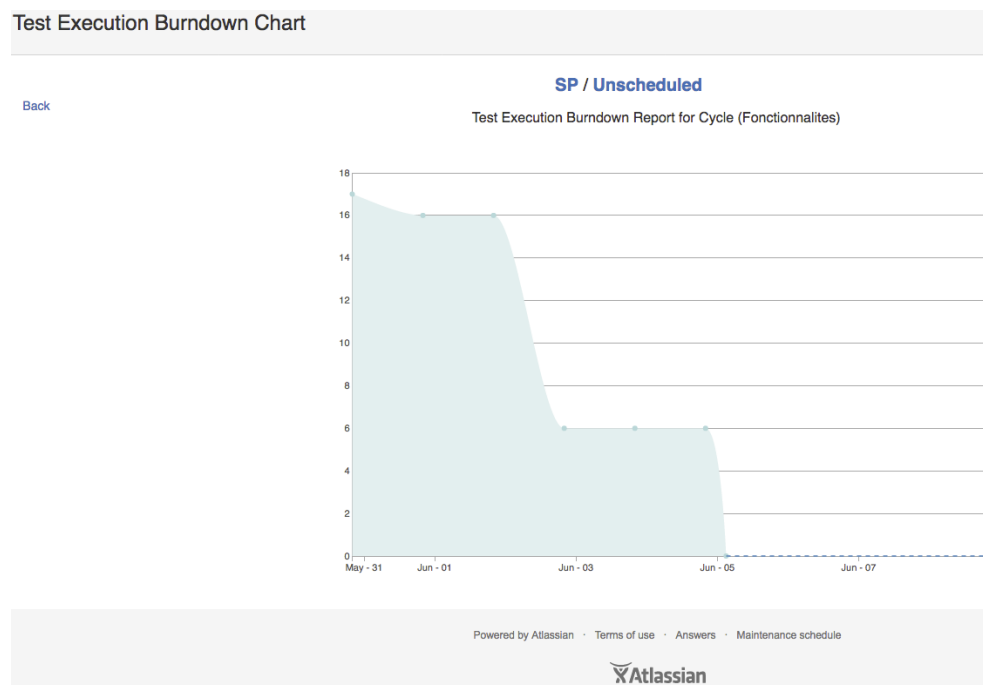
- Plusieurs tests AdHoc n'ont jamais été effectués, expliquez quelle pourrait en être la cause, étant donné qu'ils étaient planifiés de façon permanente à chaque sprint.
- 1 seul test de régression n'a jamais été exécuté jusqu'à maintenant. En comparant avec les statistiques de vélocité, expliquez pourquoi.
- Plusieurs tests de fonctionnalité n'ont jamais été exécutés jusqu'à maintenant. En comparant avec les statistiques de vélocité, expliquez pourquoi.
- Les tests de régression comptent quelques échecs. Justifiez en quoi l'équipe de développement qui produit plus d'incrément que prévu pourrait produire plus d'échec que prévu lors de l'exécution des tests.

Le [graphique d'exécution des tests par cycle de tests](#), quant à lui, nous montre les statistiques quant au nombre de tests effectués au total par cycle (type) de test.



- e. Afin de préparer une suggestion de changement à la planification des cycles tests, expliquez pourquoi vous croyez qu'il faudrait corriger la situation actuelle.

Un autre graphique, le graphique de [Burndown d'exécution](#) nous montre la répartition de la charge du travail dans le temps, pour le sprint actuel, pour les tests fonctionnels;



Le graphique nous montre que la journée de travail du 1^{er} juin 2017 aucun test n'a été exécuté, et que plus de la moitié des tests ont été exécutés la journée du 2 juin 2017.

Également, aucun test n'a été exécuté entre le 3 et le 5 juin 2017, en raison du weekend. Par contre, au retour au travail le lundi, tous les tests planifiés restants ont été exécutés. Les testeurs sont donc sans travail pour le restant du sprint (le sprint actuel)

- f. Faut-il obligatoirement ajouter des tests fonctionnels pour s'assurer qu'ils exécutent des tests 100% de leur temps de travail? Pourquoi?
- g. Si la courbe nous montre l'exécution de tous les tests fonctionnels terminés avant la fin du sprint, pourquoi cela est-il correct dans cette entreprise?
- h. Si la courbe montre l'exécution de plusieurs tests fonctionnels chaque jour, pendant plusieurs jours, à un rythme constant (plus ou moins le même nombre de tests chaque jour), cela est-il conforme? Pourquoi?

3. Gestion des défauts

La table de catégorisation des défauts utilisée à l'évaluation 2 prévaut dans votre entreprise;

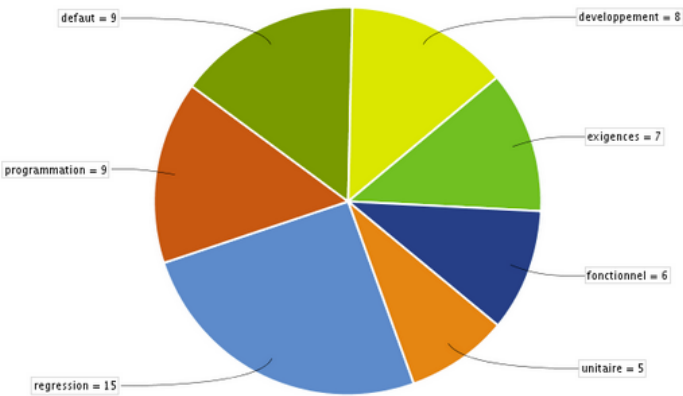
Catégories	Valeurs possibles
Type	Erreur
	Défaut
Domaine	Développement
	Tests
Phase du développement	Exigences
	Conception
	Implémentation
	Programmation
	Tests
Niveau de test	Unitaire
	Fonctionnel
	Acceptation

Le [rapport personnalisé en fonction des étiquettes des défauts](#) (Labels) vous fournit les informations suivantes :

Pie Chart Report

Project: [Evaluation Finale - Bibliotheque](#) (Labels)

Chart



Data Table

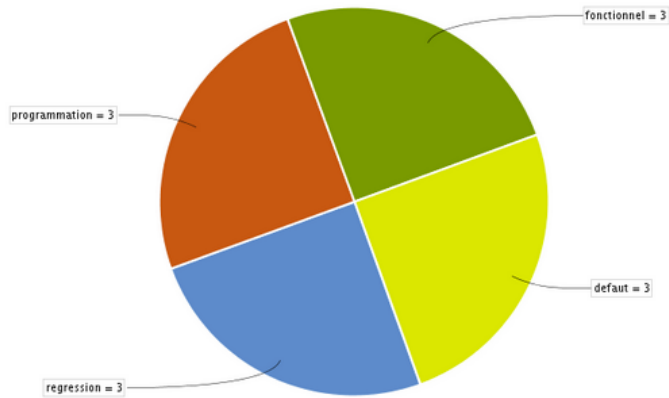
	Issues	%
regression	15	30%
programmation	9	18%
default	9	18%
developpement	8	16%
exigences	7	14%
fonctionnel	6	12%
unitaire	5	10%

Vous cherchez donc à expliquer le pourcentage de défaut aussi élevé dans les tests de régression, puisque si ceux-ci ont été exécutés 1 fois avec succès, ils devraient toujours être exécutés avec succès par la suite. Voici ce que dit le même rapport en ce qui concerne uniquement les défauts de type régression :

Pie Chart Report

Filter: [default](#)[Fonctionnel](#)[Prog](#) (Labels)

Chart



Data Table

	Issues	%
regression	3	100%
programmation	3	100%
fonctionnel	3	100%
default	3	100%

Powered by Atlassian · [Terms of use](#) · [Answers](#) · [Maintenance schedule](#)



- Quelle conclusions pouvez-vous tirer sur le type de défauts qui affectent le projet jusqu'à maintenant?
- Quelle est la cause principale des défauts répertoriés?
- Quelle suggestions pourriez-vous faire afin d'améliorer la situation?