

Synthlab : Rapport d'organisation

Auteurs :
ANNE Aurélien,
COISNARD-SIMON Marie,
FORTUN Nicolas,
LAURENT Julien,
NOMANE Ahmed
SALMON Kevin

7 mars 2018

Table des matières

1	Introduction	1
2	Infrastructure mise en place	1
2.1	Intégration continue	1
2.2	Gestion Agile	2
2.3	Communication	2
2.4	Déroulement d'un sprint	2
3	Sprint 1	2
3.1	Backlog	2
3.2	Déroulement	3
3.3	Sprint review	3
4	Sprint 2	4
4.1	Backlog	4
4.2	Déroulement	5
4.3	Sprint review	6
5	Sprint 3	6
5.1	Backlog	7
5.2	Déroulement	7
5.3	Sprint review	9
6	Conclusion et retour critique	9

1 Introduction

Notre groupe est composé de 6 étudiants de Master 2 IL. Le but de ce projet est de créer un synthétiseur numérique permettant de manipuler des flux audio via différents modules. Ce rapport présente l'organisation et l'infrastructure mises en place afin de permettre le bon déroulement du projet.

2 Infrastructure mise en place

2.1 Intégration continue

Nous avons mis en place une VM sur les serveurs de l'Istic avec un Jenkins, SonarQube et Phabricator pour la revue de code. Chaque logiciel été lancé dans un container Docker.

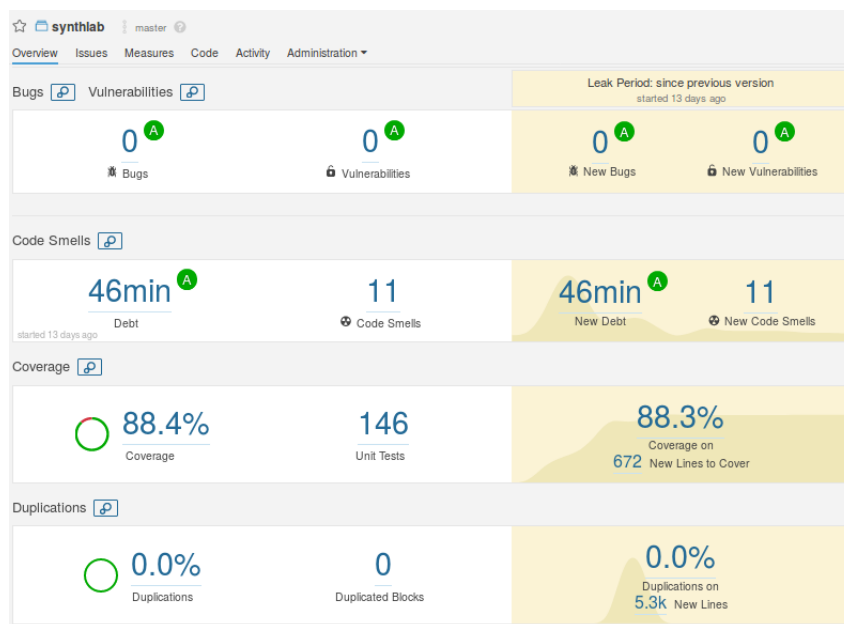


FIGURE 2 – Contrôle de qualité de code grâce à Sonar

Les sources du projet sont sur Github. À chaque commit, une série d'étapes spécifiques s'activent :

- Récupération du commit sur Phabricator ;
- Lancement du build sur Jenkins ;
- Lancement des tests sur Jenkins ;
- Lancement de l'analyse de qualité avec SonarQube.

Si Jenkins rencontre un problème lors de la compilation, un mail est envoyé à l'équipe. Phabricator, quant à lui, nous permet de suivre la relecture du code. Lorsqu'un développeur valide ou commente un commit, ou bien soulève un problème, un mail est directement envoyé à l'équipe.

S	M	Nom du projet ↓	Dernier succès	Dernier échec	Dernière durée
		synthlab	11 mn - #232	4 h 47 mn - #220	28 s

FIGURE 3 – Intégration continue avec Jenkins

2.2 Gestion Agile

Nous avons eu 3 semaines pour réaliser ce projet. Nous avons donc réalisé trois sprints d'une semaine chacun. Pour le suivi de nos users stories (US), nous utilisons Taiga. Cet outil nous permet de suivre en temps réel l'avancement du sprint. Les US sont divisées en tâche. Cependant, il ne permet pas d'affecter de complexité, ou plusieurs personnes aux tâches. Nous avons utilisé en parallèle un fichier Excel pour effectuer le suivi du développement, afin de pallier ce manque.

2.3 Communication

Nous utilisons Slack pour communiquer. Nous avons créé un channel entre les membres de l'équipe et un autre avec le product owner pour lui poser des questions si besoin.

2.4 Déroulement d'un sprint

Un sprint se déroule de la manière suivante :

- Sprint planning avec le Scrum Master le lundi après-midi. Le but est de définir et d'évaluer les US à réaliser durant le sprint. Les points de complexité sont définis par un planning poker ;
- Découpage en tâches des US avec l'équipe ;
- Distribution des tâches entre développeurs ;
- Daily meeting tous les jours à 13h ;
- À la fin du sprint, démonstration avec le product owner et rétrospective avec le scrum master.

3 Sprint 1

Le sprint 1 se déroulait du 30 janvier au 5 février 2018. L'objectif de ce sprint était d'obtenir un son sur la sortie son d'un PC, produit par un module VCO, réglé à 1 kHz et avec une forme d'onde carrée, connecté au module de sortie.

3.1 Backlog

Pour ce premier sprint, nous avons pris 6 US pour un total de 4800 points de valeur métier. Nous avons estimé notre capacité à 78 avec un point de complexité arbitraire valant 2 heures.

En plus des US, nous avons estimé à 3 points de complexité le temps nécessaire à la mise en place de notre infrastructure de projet. Nous avons également mis en place de la revue de code, et pris en compte du temps pour réaliser le suivi et la release. Enfin, nous avons gardé 6 points de complexité en cas de problème.

Étant une équipe de 6 personnes et ayant des tâches bloquantes, car peu nombreuses au début du projet, nous avons travaillé par binômes afin d'avancer plus vite dans nos développements et pour que tout le monde ait quelque chose à faire. Nous avons globalement réparti les tâches en fonction de l'infrastructure, la partie graphique et l'implémentation des modules.

Pour la partie graphique, nous avons négocié une US supplémentaire dont l'objectif est de rendre l'interface visuellement agréable, tant pour les câbles que pour les modules ou le board. Ceci nous a permis de faire des visuels simplistes pour rapidement se focaliser sur les tâches importantes de la gestion audio.

Usecase	Valeur	Compris (O/N)	Détailé (O/N)	Réalisable (O/N)	Date évaluation n DoR	Read y ?	Priorisé	Complexité
Evaluation du travail par un rapport.txt	1000	O	O	O	30-janv.	O		2
Evaluation du travail par une soutenance	1000	O	O	O	30-janv.	O		2
Création de connexions	500	O	O	O	30-janv.	O	1	5
Création de module de sortie	500	O	O	O	30-janv.	O	1	8
Création de module VCO type A	500	O	O	O	30-janv.	O	1	20
Création de plan de montage global (#GLOB)	500	O	O	O	30-janv.	O	1	5
Manipulation de signaux	500	O	O	O	30-janv.	O		3
Création de module VCA	300	O	O	O	30-janv.	O	2	13

FIGURE 4 – Backlog du sprint 1

3.2 Déroulement

Durant ce premier sprint, la mise en place de l'infrastructure a pris un peu plus de temps que prévu car la mise en place de Phabricator a été compliquée. De plus, notre VM hébergée à l'Istic a rencontré un problème, nous avons donc dû transférer notre infrastructure sur un serveur personnel le temps que cela revienne à la normale.

Phabricator ayant été mis en place tardivement, nous l'avons relativement peu utilisé durant le sprint. Une fois en place, nous avons décidé de ne pas relire tous les précédents commits puisqu'ils étaient déjà très nombreux et que la structure de notre projet avait changé depuis. Nous avons donc décidé arbitrairement d'un certain commit à partir duquel nous commençons à faire la relecture des nouveaux commits.

Durant ce sprint, nous avons dû réfléchir à la structure de nos classes. Cela s'est fait par réflexion de 2 ou 3 personnes sur les différents aspects du projet. Nous avons par exemple repris tous les modules du projet pour repérer quels seraient leurs éventuels points communs et différences en terme de ports et fonctionnalités. Il a également fallu adapter notre vision à JSyn afin d'utiliser la librairie efficacement.

3.3 Sprint review

Lors de la sprint review, nous avons présenté l'état courant du projet. C'est-à-dire des modules fonctionnels pouvant être reliés en ligne de commande, une interface graphique où chaque module est lié à sa version graphique mais avec l'impossibilité de les relier. En effet, nous avons sous-estimé la réalisation des câbles reliant visuellement les modules. De ce fait, nous n'avons donc pas pu valider l'objectif de ce premier sprint.

Nous avons donc pu dégager les points suivants :

- La partie graphique reste à travailler (câbles, etc)
- La partie graphique et la partie son ne sont pas liés
- Les modèles du menu ne sont pas des prototypes du module cible, ils doivent être peu encombrants et non spécialisés avant l'ajout sur le board
- F0 sur le VCO est une constante applicative et non pas un réglage du VCO

- il manque le label pour l'atténuation du VCA. L'atténuation pourrait être un slider avec des chiffres négatifs et un écart type plus grand
- Pas de livraison
- Soutenance et rapport non soumis
- Sonar et Jenkins en place, tests partiellement en place
- Phabricator pour la relecture de code

Les aspects positifs et négatifs relevés lors de ce premier sprint sont :

Positif	Négatif
Pair programming : relecture, réflexion efficace	Adhérences nombreuses entre tâches
Slack, temps de réponse du PO bon	Réservation des salles
Usecases bien clairs	Personnes absentes pour maladie
Bonne équipe et bonne répartition	Démarrage avec JavaFX compliqué
Vision partagée via Taiga	Pas de complexité ni de partage sur les tâches Taiga, uniquement l'avancement des US
Solutions techniques adaptée	Jira payant
Structure VM ok	Les titres des tâches ne sont parfois pas suffisant : elles sont à détailler
Tableaux blancs	

Nous avons dégagé quelques points d'améliorations avec les côtés négatifs : Commenter les tâches nécessaires, utiliser un fichier Excel en plus de Taiga pour gérer la complexité et les affectations aux tâches.

4 Sprint 2

L'objectif du sprint n°2 est de pouvoir réaliser le montage suivant :

- un VCO n°1 réglé à la fréquence de 1 Hz (comme un LFO) ;
- un VCO n°2 réglé à la fréquence de base de 1 kHz ;
- un VCA ;
- un EG ;
- un module de sortie son.

Câblage :

- la sortie out du VCO n°1 est reliée à l'entrée fm du VCO n°2 ;
- la sortie out du VCO n°2 est reliée à l'entrée in du VCA ;
- la sortie out du VCA est reliée à l'entrée in du module de sortie son.
- la sortie out du VCO n°1 est reliée à l'entrée gatede l'EG ;
- la sortie out de l'EG est reliée à l'entrée am du VCA.

4.1 Backlog

La valeur estimée en début de sprint était de 5900 points valeur métier et notre complexité estimée en début de sprint est de 114.

N'ayant pas validé l'objectif du sprint n°1, nous avons reporté toutes ses US dans le sprint n°2. Nous avons estimé leur complexité à 27. Pour les autres US, nous avons estimé le total de leur complexité à 85. Nous avons aussi pris celles nécessaires à l'objectif du sprint n°3, ainsi que quelques autres US, car nous avons la place de les prendre. En dehors des US, nous avons également prévu les points de complexité suivants :

- 8 Risque

- 4 Release
- 4 Suivi
- 2 Intégration continue
- 8 Revue

Ce qui nous fait un total de capacité de réalisation de 88 et 3 de marge.

Usecase	Valeur	Compris (O/N)	Détaillé (O/N)	Réalisabl e (O/N)	Date évaluation DoR	Ready ?	Prioris é	Compl exité
Evaluation du travail par un rapport	1000	O	O	O	30-janv.	O	1	2
Evaluation du travail par une soutenance	1000	O	O	O	30-janv.	O	1	2
Manipulation de signaux	500	O	O	O	30-janv.	O		2
Création de module VCA	300	O	O	O	30-janv.	O	2	5
Création de connexions	500	O	O	O	30-janv.	O	1	8
Création de module de sortie	500	O	O	O	30-janv.	O	1	2
Création de module VCO type A	500	O	O	O	30-janv.	O	1	2
Création de plan de montage global (#GLOB)	500	O	O	O	30-janv.	O	1	8
Création de module VCF type LP	300	O	O	O	30-janv.	O		10
Création de module EG	200	O	O	O	30-janv.	O		13
Création de module oscilloscope	200	O	O	O	30-janv.	O		16
Création de module réplicateur	200	O	O	O	30-janv.	O		5
Création de module VCF type HP (passe-haut)	200	O	O	O	30-janv.	O		10

FIGURE 5 – Backlog Sprint 2

4.2 Déroulement

Nous avons réparti les personnes entre les tâches permettant de clôturer le sprint n°1 et celles faisant partie de l'objectif du sprint n°2. Le deuxième jour du sprint, nous avons pu terminer la connexion des câbles. Ensuite, nous avons pu vérifier l'intégration des modules dans le board, et effectuer les corrections et ajustements nécessaires. Les modules du sprint n°2 ayant bien avancés entre temps, nous avons pu commencer à travailler sur les US suivantes telles que celles concernant le VCF passe-bas et l'oscilloscope.

Le jeudi 8 février, nous estimions avoir une version beta pouvant valider le sprint 2. De plus, nous avons réalisé la quasi totalité des US et tâches prévues pour ce sprint. Nous avons donc contacté le PO le lendemain après notre daily pour avoir un retour. Nous avons intégré au fur et à mesure des US supplémentaires concernant le sprint n°3, puis des US optionnelles.

Nous avons donc ajouté en cours de route les US suivantes :

- Sauvegarde et restauration de configuration ;
- Sauvegarde du son produit dans un fichier son au format WAV ou MP3 (#MP3) ;
- Aspect visuel des connexions ;
- Changement d'aspect IHM (#SKIN).

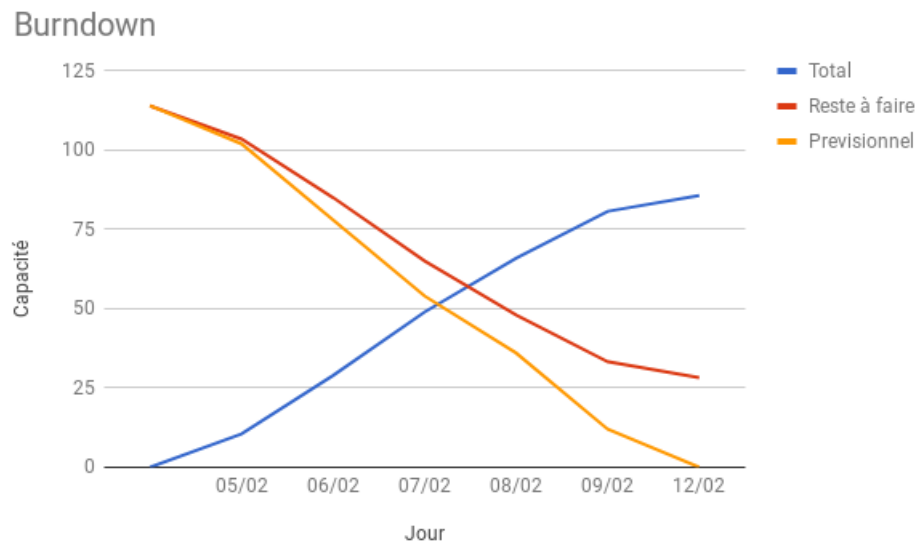


FIGURE 6 – Burndown Sprint 2

4.3 Sprint review

Durant notre démo, nous avons d’abord présenté les réalisations correspondant au backlog initial du sprint n°2. Pour les US qui n’étaient pas prévues, nous avons présenté uniquement celles étant terminées telles que le changement de skin, l’enregistrement au format WAV ainsi que l’aspect visuel des connexions. Nous avons donc eu les remarques suivantes :

- Octave du VCO ne marche pas, seulement réglage fin
- Positionnement sur le board, modules trop grands ?
- Défaut d’affichage oscilloscope
- Latence sur oscilloscope / sur signal
- Decay ne fonctionne pas sur EG (ou pas à l’échelle)
- Release ne fonctionne pas sur EG (ou pas à l’échelle)
- Atténuateur VCA ne fonctionne pas
- Manque de lisibilité des labels sur le skin métal

A l’issu de cette review, nous avons également revu les aspect positifs et négatifs durant le sprint. Nous avons voté pour désigner les plus importants et avons obtenu ce tableau :

Positif	Négatif
Implémentation techniques des modules Infrastructure en place efficace (Jenkins, Sonar, Phabricator) PO présent	Saisie activité excel JSyn (documentation faible, exemples pas clairs, refactor) Difficultés de choix pour alimenter le sprint Frustration Scrum Master de ne pas pouvoir être plus présent Suivi agile insuffisant avec les outils

5 Sprint 3

L’objectif du sprint n°3 est de pouvoir réaliser les deux montages A et B suivants. Les composants du montage A sont :

- un VCO n°1 réglé à la fréquence de 0,5 Hz (comme un LFO) ;
- un VCO n°2 réglé à la fréquence de base de 1 kHz, signal dent de scie ;
- un VCF LP 24 dB/octave réglé à la fréquence de 1 kHz
- un VCA ;
- un EG ;
- un module de sortie son.

Câblage du montage A :

- la sortie out du VCO n°1 est reliée à l'entrée fm du VCF ;
- la sortie out du VCO n°2 est reliée à l'entrée in du VCF ;
- la sortie out du VCF est reliée à l'entrée in du VCA ;
- la sortie out du VCA est reliée à l'entrée in du module de sortie son ;
- la sortie out du VCO n°1 est reliée à l'entrée gatede l'EG ;
- la sortie out de l'EG est reliée à l'entrée am du VCA.

Les composants du montage B sont :

- un VCO n°1 réglé à la fréquence de 1,5 kHz, signal triangle ;
- un VCO n°2 réglé à la fréquence de base de 1 kHz, signal triangle ;
- un réplicateur de ports ;
- un module de sortie son.

Câblage du montage B :

- la sortie out du VCO n°1 est reliée à l'entrée fm du VCO n°2 ;
- la sortie out du VCO n°2 est reliée à l'entrée in du réplicateur ;
- la sortie out1 du réplicateur est reliée à l'entrée fm du VCO n°1 ;
- la sortie out2 du réplicateur est reliée à l'entrée in du module de sortie son.

Les configurations seront préparées à l'avance et chargées à partir de leurs sauvegardes.

5.1 Backlog

La valeur estimée en début de sprint était de 3780 points de valeur métier et notre complexité estimée en début de sprint est de 114.

Nous avons donc 87 points de complexité pour toutes les US ainsi que :

- 8 Risque
- 4 Release
- 4 Suivi
- 2 Intégration continue
- 16 Revue

Use case	Valeur	Compris (O/N)	Détaillé (O/N)	Réalisable (O/N)	Date évaluation DoR	Ready ?	Priorité	Complexité
Evaluation du travail par un rapport	1000	O	O	O	30-janv	O		10
Evaluation du travail par une soutenance	1000	O	O	O	30-janv	O		10
Création de module EG	200	O	O	O	30-janv	O		5
Création de module oscilloscope	200	O	O	O	30-janv	O		5
Changement d'aspect IHM (#5K2N)	150	O	O	O	30-janv	O		2
Création de module VCF type HP (passe-haut)	200	O	O	O	30-janv	O		2
Création de module bruit blanc	80	O	O	O	30-janv	O		3
Création de boucles	300	O	O	O	30-janv	O		1
Création de module mixer	200	O	O	O	30-janv	O		8
Création de module séquenceur	200	O	O	O	30-janv	O		13
Clavier de commande (#KEYB)	150	O	O	O	30-janv	O		8
Sauvegarde et restauration de configuration	100	O	O	O	30-janv	O		20

FIGURE 7 – Backlog Sprint 3

5.2 Déroulement

Après avoir divisé les US en tâches, nous avons travaillé en parallèle à la résolution des remarques évoquées lors de la review du sprint (oscilloscope, module EG et aspect IHM) et au

développement des nouvelles fonctionnalités. Il n'y a pas eu de grande difficultés pour le clavier de commande, séquenceur, mixer, VCFHP et module de bruit blanc.

La sauvegarde et restauration de l'espace de travail a été la plus complexe. Tout d'abord, nous avons essayé de sauvegarder directement les modules avec la librairie Jackson. Malheureusement, JSyn ne le permettait pas car avec les connexions entre modules, il y avait des références circulaires. Ainsi, Jackson ne pouvait pas générer un fichier de sauvegarde. Il fallait donc réaliser une sauvegarde "à la main". Nous avons défini un format de données, puis réalisé la sauvegarde des modules. Le chargement était également compliqué car il fallait recréer les modules puis leurs interactions à la fois sur l'IHM et de manière fonctionnelle.

Avec l'avance acquise au sprint 2, l'objectif du sprint 3 était atteint vers les 70% du sprint. Nous avons fait appel au PO pour qu'il puisse nous dire si le produit lui convenait. C'était le cas sans pour autant valider le sprint 3 avant le sprint review.

Nous avons pu profiter de cette avance pour travailler les 4 rapports, corriger des bugs mineurs, améliorer la fiabilité des traitements et faire de la revue de code.

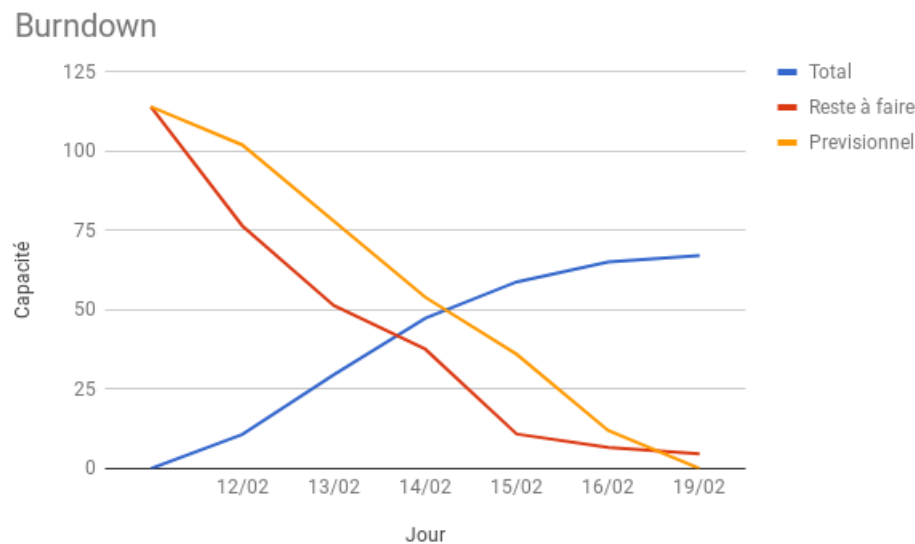


FIGURE 8 – Burndown Sprint 3

5.3 Sprint review

Les aspects positifs et négatifs relevés lors de ce dernier sprint sont :

Positif	Négatif
En avance donc plus de temps pour les rapports	Instabilité de la VM
Bon rapport Sonar : 90% de couverture de code, 0 code smells, 0 bugs	Parties du code non testables automatiquement (Controller)
Suivi Excel efficace	Taiga : obligé de faire un Excel
US bien taillées pour 3 semaines	Salles instables, changement de bâtiment
Produit fini (on peut jouer du Bach), fonctionne et conforme	Pas de créneau après le sprint review
Agile : PO disponible, bonne communication par Slack, réponse aux questions rapide	
Scrum Master : équipe sympa, projet rigolo	
Progression métier de l'équipe	

6 Conclusion et retour critique

Nous avons choisi de réaliser cette architecture, au vu de la courte durée du projet (trois semaines). Il est ainsi simple et rapide d'étendre l'application en ajoutant de nouveaux modules, et les bugs d'un module n'auront aucun impact sur le fonctionnement des autres modules.

Ce projet nous a introduit aux méthodes Agiles, notamment SCRUM. Nous avons appris à travailler en équipe, découper un *use case* en plus petites tâches, et à estimer la complexité d'ajout de fonctionnalités.