

Liens utiles

Vous trouverez dans ces liens utiles des lectures et des outils qui n'ont pas pu être présentés dans le cours.

Lectures conseillées

- The Architecture of Open Source Applications [<http://www.aosabook.org/en/index.html>] (eBook)
- 10 règles de la NASA pour développer un programme critique [<http://sdtimes.com/nasas-10-rules-developing-safety-critical-code/>] (en C)
- Lessons From a Decade of IT Failures [<http://spectrum.ieee.org/static/lessons-from-a-decade-of-it-failures>]
- Technical Development Guide [<https://www.google.com/about/careers/students/guide-to-technical-development.html>] (Google Careers)
- Programming Languages | Hammer Principle [<http://hammerprinciple.com/therighttool>] (Ou : comment bien choisir son langage de programmation ?)
- Software disenchantment [<http://tonsky.me/blog/disenchantment/>]
 - Le désenchantement du logiciel [<https://blog.romainfallet.fr/desenchantment-logiciel/>] (traduction)
- C'est la faute du développeur [<https://www.jesuisundev.com/la-faute-du-developpeur/>]
- How often do people actually copy and paste from Stack Overflow ? Now we know. [<https://stackoverflow.blog/2021/04/19/how-often-do-people-actually-copy-and-paste-from-stack-overflow-now-we-know/>]
- Best practices for writing code comments [<https://stackoverflow.blog/2021/12/23/best-practices-for-writing-code-comments/>]

Planification

- GanttProject [<http://www.ganttproject.biz/>]
- Community LibrePlan [<http://www.libreplan.com/>]
- TaskJuggler [<http://www.taskjuggler.org/>]
- Focalboard: Open source alternative to Trello, Asana, and Notion [<https://www.focalboard.com/>]

Conception

Outils de modélisation UML (Open-source)

- Modelio [<https://www.modelio.org>]
- Umbrello [<http://uml.sourceforge.net/>]
- ArgoUML [<http://argouml.tigris.org/>]
- StarUML [<http://staruml.sourceforge.net/en/>]

- [UML Graph \[http://www.umlgraph.org/\]](http://www.umlgraph.org/) est un outil pour représenter automatiquement des diagrammes UML

Patrons de conception (Design Patterns)

- [Patrons de conception \(avec implémentation en Java\) \[http://rpouiller.developpez.com/tutoriel/java/design-patterns-gang-of-four/\]](http://rpouiller.developpez.com/tutoriel/java/design-patterns-gang-of-four/)
- [Discussion au sujet du patron de conception MVC et des alternatives possibles \[https://www.infoq.com/fr/articles/no-more-mvc-frameworks\]](https://www.infoq.com/fr/articles/no-more-mvc-frameworks)

Implémentation

Templates

- [cookiecutter \[https://github.com/audreyr/cookiecutter\]](https://github.com/audreyr/cookiecutter) est une collection de templates pour des projets dans différents langages sur différentes plateformes; ils permettent d'avoir rapidement un environnement opérationnel.

Cours de Java

- [Développons en Java \[http://www.jmdoudoux.fr/accueil_java.htm\]](http://www.jmdoudoux.fr/accueil_java.htm) de Jean-Michel Doudoux
- [Développons en Java avec Eclipse \[http://www.jmdoudoux.fr/accueil_java.htm\]](http://www.jmdoudoux.fr/accueil_java.htm) de Jean-Michel Doudoux

Journalisation

- [Apache logging services \[http://logging.apache.org\]](http://logging.apache.org) regroupe plusieurs projets open-source permettant la journalisation du comportement d'applications (langages supportés : Java, PHP, C++, .Net)
 - [Log4j \[http://logging.apache.org/log4j/\]](http://logging.apache.org/log4j/) (a logging library for Java) permet de journaliser l'activité d'une application Java.
 - [Chainsaw \[http://logging.apache.org/chainsaw/\]](http://logging.apache.org/chainsaw/) permet de visualiser graphiquement les log obtenus avec log4j.

Environnement de développement intégré

- [Eclipse \[https://www.eclipse.org/\]](https://www.eclipse.org/) est un IDE polyvalent grâce à l'utilisation de plugins.
- [Netbeans \[https://netbeans.org/\]](https://netbeans.org/) est une bonne alternative (Principaux langages supportés : Java, HTML5, PHP, C/C++)

Java

C/C++

- [Code::Blocks \[http://www.codeblocks.org/\]](http://www.codeblocks.org/)

Python

* [Pycharm \[http://www.pycharm.org/\]](http://www.pycharm.org/) (Il y a une licence pour l'université)

PHP

- [PDT \[http://www.eclipse.org/pdt/\]](http://www.eclipse.org/pdt/) (PHP Development Tools)
- [Symfony2 Eclipse plugin \[http://symfony.dubture.com/\]](http://symfony.dubture.com/) (permet d'intégrer les spécificités de Symfony2)

- [Tutoriel d'installation du plugin \[http://www.lafermeduweb.net/tutorial/symfony2-eclipse-integrer-le-framework-symfony-2-a-eclipse-pour-mieux-developper-48.html\]](http://www.lafermeduweb.net/tutorial/symfony2-eclipse-integrer-le-framework-symfony-2-a-eclipse-pour-mieux-developper-48.html)
- Mémo
 - [Pour utiliser Symfony2 avec PostgreSQL \[http://stackoverflow.com/questions/10588646/how-to-change-a-database-to-postgresql-with-symfony-2-0\]](http://stackoverflow.com/questions/10588646/how-to-change-a-database-to-postgresql-with-symfony-2-0)
 - [Packages d'installation de PHP 5.x pour OSX \[http://php-osx.liip.ch/\]](http://php-osx.liip.ch/)

Frameworks

PHP

- [Phalcon \[http://phalconphp.com\]](http://phalconphp.com) est un micro-framework PHP programmé en C, c'est le plus rapide à ce jour en termes de requêtes par seconde (source [\[http://systemsarchitect.net/performance-benchmark-of-popular-php-frameworks/\]](http://systemsarchitect.net/performance-benchmark-of-popular-php-frameworks/))
- [Symfony2 \[http://symfony.com/\]](http://symfony.com/) est le framework le plus populaire, il facilite grandement le développement d'applications web complexes; il est modulaire et dispose de nombreux plugins.

Python

- [Django \[https://www.djangoproject.com/\]](https://www.djangoproject.com/)

Tests

- 80 ou 90% de couverture de tests pour un nouveau projet? [\[https://blog.ippon.fr/2019/07/22/80-ou-90-de-couverture-de-tests/\]](https://blog.ippon.fr/2019/07/22/80-ou-90-de-couverture-de-tests/)

Outils de tests

JUnit

- Site officiel de JUnit [\[http://www.junit.org/\]](http://www.junit.org/)
- [JUnit \[http://www.jmdoudoux.fr/java/dej/chap078.htm\]](http://www.jmdoudoux.fr/java/dej/chap078.htm) - Chapitre de *Développons en Java* de Jean-Michel Doudoux
- [JUnit \[http://www.jmdoudoux.fr/java/dejae/chap011.htm\]](http://www.jmdoudoux.fr/java/dejae/chap011.htm) - Chapitre de *Développons en Java avec Eclipse* de Jean-Michel Doudoux
- [JUnitum \[http://code.google.com/p/junitum/\]](http://code.google.com/p/junitum/) est une extension de JUnit basé sur les idées développées dans TestNG

TestNG

- Site officiel de TestNG [\[http://www.testng.org/\]](http://www.testng.org/)
- [TestNG \[http://www.spalud.info/testng/\]](http://www.spalud.info/testng/) (en français)

PostgreSQL

- [pgTAP: Unit Testing for PostgreSQL \[https://pgtap.org/\]](https://pgtap.org/)
 - [Écrire ses tests unitaires en SQL \[https://fljd.in/2020/05/14/ecrire-ses-tests-unitaires-en-sql/\]](https://fljd.in/2020/05/14/ecrire-ses-tests-unitaires-en-sql/)

Python

- [Tox \[https://tox.readthedocs.io/\]](https://tox.readthedocs.io/) vise à standardiser et à automatiser les tests en Python

Quick Perf

- [quick-perf \[https://github.com/quick-perf\]](https://github.com/quick-perf) est une librairie Java pour évaluer et améliorer les performances d'une application.

Web

- [gremlins.js \[https://github.com/marmelab/gremlins.js\]](https://github.com/marmelab/gremlins.js) est un application pour tester de manière exhaustive les applications web et node.js

Frameworks de test

Behaviour Driven Development

- [Cucumber \[https://cucumber.io/\]](https://cucumber.io/) est un outil d'automatisation des tests basé sur les spécifications du comportement attendu (BDD)
 - [Calabash / Cucumber - Ecrire des tests fonctionnels pour Mobile \[https://blog.eleven-labs.com/fr/calabash-cucumber-ecrire-des-tests-fonctionnels-pour-mobile/\]](https://blog.eleven-labs.com/fr/calabash-cucumber-ecrire-des-tests-fonctionnels-pour-mobile/) (iOS)

JSystem

- [JSystem \[http://www.jssystemtest.org/\]](http://www.jssystemtest.org/) est un framework reconnu conçu pour écrire et gérer graphiquement des tests d'applications. Il est basé sur JUnit et ANT.

STAF

- [Software Testing Automation Framework\(STAF\) \[http://staf.sourceforge.net/index.php\]](http://staf.sourceforge.net/index.php)

GUITAR

- [GUI Testing frAmewoRk \[http://guitar.sourceforge.net/\]](http://guitar.sourceforge.net/)

Robot Framework

- [Robot Framework \[https://robotframework.org/\]](https://robotframework.org/)
- [Comprendre Robot Framework en 5 minutes \[https://www.gologic.ca/robot-import/\]](https://www.gologic.ca/robot-import/)

Web

- [zuul \[https://github.com/defunctzombie/zuul\]](https://github.com/defunctzombie/zuul) permet de tester du code javascript dans plusieurs navigateurs web simultanément.

Tests de montée en charge serveur web

- [Gatling \[https://gatling.io/\]](https://gatling.io/)
- [Liste d'outils pour tester la rapidité d'un site web \[http://bertiaux.fr/2011/12/les-7-meilleurs-outils-pour-tester-la-rapidite-de-votre-site-web/\]](http://bertiaux.fr/2011/12/les-7-meilleurs-outils-pour-tester-la-rapidite-de-votre-site-web/)

Listes d'outils de test

- Liste d'outils pour des tests fonctionnels (open-source) [<http://www.opensourcetesting.org/functional.php>]
- Liste d'outils pour des tests fonctionnels (open-source) [<http://www.functionaltestingtools.com/?opensource>]
- Liste d'outils pour tester des GUI (wikipédia) [http://en.wikipedia.org/wiki/List_of_GUI_testing_tools]
- Liste d'outils en Java pour tester des sites web [<http://java-source.net/open-source/web-testing-tools>]

Outils de gestion (version,projet,etc.)

Outils de base

- Subversion [<http://subversion.apache.org/>]
- Git [<http://git-scm.com/>]
- Mercurial [<http://mercurial.selenic.com/>]
- Gitea [<https://gitea.io/en-us/>]

Forges logicielles

- Bazaar [<http://bazaar.canonical.com/en/>]
- ChiliProject [<https://www.chiliproject.org/>] (fork Redmine)
- LibreSource [<http://dev.libresource.org/>]
- Maven [<http://maven.apache.org/>]
- Redmine [<http://www.redmine.org>]
- Trac [<http://trac.edgewall.org/>]
- Tuleap [<https://tuleap.net/>]

Integration continue

- Jenkins [<http://jenkins-ci.org>] est un outil d'intégration continue; il permet d'automatiser et de répéter à intervalles réguliers les tâches liées aux activités récurrentes d'un projet (la construction, les tests de la version la plus récente l'application ...)

Visualisation

- Gource [<http://code.google.com/p/gource/>] permet de visualiser l'évolution d'un projet.

Collaboration

- Bit: Easily share code between projects with your team [<https://github.com/teambit/bit>]
- Tracim [<https://www.tracim.fr/>]

Gestion de tâches

- Request Tracker [<https://bestpractical.com/request-tracker>] est un système de gestion de tickets qui permet de personnaliser les workflows
- Original | TaskFreak [<http://www.taskfreak.com/original>] est un gestionnaire de tâche inspiré par l'approche GTD.

Qualité

Articles

- [Bazman's testing pages \[http://bazman.tripod.com/index.html\]](http://bazman.tripod.com/index.html) contient plusieurs articles (et de nombreux conseils pratiques) pour la mise en oeuvre de tests. Par exemple : [GUI Testing Checklist \[http://bazman.tripod.com/checklist.html?button1=GUI+Testing+Checklist\]](http://bazman.tripod.com/checklist.html?button1=GUI+Testing+Checklist) malgré son âge, cet article est particulièrement exhaustif en ce qui concerne la réalisation des tests d'une interface utilisateur graphique.
- [Software Development Quality Assurance Zone \[http://www.sqazone.net/\]](http://www.sqazone.net/) est un site spécialisé qui contient de nombreuses ressources (articles, conseils, liens, outils, ...) concernant la qualité du logiciel.
- [Applications : le « mauvais code » coûte très cher aux entreprises \[https://www.silicon.fr/developpeurs-mauvais-code-coute-entreprises-220059.html\]](https://www.silicon.fr/developpeurs-mauvais-code-coute-entreprises-220059.html)

Gestion de projets

- [Scrum didn't work for my startup — so I designed an alternative project management method \[https://thenextweb.com/growth-quarters/2020/07/01/scrum-didnt-work-for-my-startup-so-i-designed-an-alternative-project-management-method-syndication/\]](https://thenextweb.com/growth-quarters/2020/07/01/scrum-didnt-work-for-my-startup-so-i-designed-an-alternative-project-management-method-syndication/)
- [Gestion de projet : 25 méthodes pour bien s'organiser \[https://www.blogdumoderateur.com/gestion-de-projet-methodes-bien-sorganiser/\]](https://www.blogdumoderateur.com/gestion-de-projet-methodes-bien-sorganiser/)

Métriques de qualité

- [Outil générant automatiquement un rapport de qualité \[http://metric-fu.rubyforge.org/\]](http://metric-fu.rubyforge.org/)
- [Sonar : un outil pour améliorer la qualité du code source d'un projet \[http://www.sonarsource.org/\]](http://www.sonarsource.org/) (permet la gestion de nombreuses métriques)
- [Software Engineering Metrics: What Do They Measure and How Do We Know ? \[http://www.kaner.com/pdfs/metrics2004.pdf\]](http://www.kaner.com/pdfs/metrics2004.pdf) (article)

Documentation

- [La documentation technique, le récit d'un échec \[https://www.damyr.fr/posts/documentation-technique-echec/\]](https://www.damyr.fr/posts/documentation-technique-echec/)
- [A practical guide to writing technical specs \[https://stackoverflow.blog/2020/04/06/a-practical-guide-to-writing-technical-specs/\]](https://stackoverflow.blog/2020/04/06/a-practical-guide-to-writing-technical-specs/)

Débogage

Méthodologie

- [How to Debug Any Problem \[https://hackernoon.com/how-to-debug-any-problem-ac6f8a867fae\]](https://hackernoon.com/how-to-debug-any-problem-ac6f8a867fae)
- [Comment déboguer un problème de manière efficace ? \[https://blog.ston3o.me/comment-debugger-un-probleme-de-maniere-efficace/\]](https://blog.ston3o.me/comment-debugger-un-probleme-de-maniere-efficace/)

Cours

- [Software Debugging - Automating the Boring Tasks \[https://www.udacity.com/course/software-debugging--cs259\]](https://www.udacity.com/course/software-debugging--cs259)

Syllabus

Lesson 1: How Debuggers work

Theory: Scientific method and its application to debugging.
 Fun fact: First bug in the history of computer science.
 Practice: Building a simple tracer.

Lesson 2: Asserting Expectations

Theory: Assertions in testing and in debugging.
 Fun fact: The most expensive bug in history.

Practice: Improving the tracer.

Lesson 3: Simplifying Failures

Theory: Strategy of simplifying failures. Binary search. Delta debugging principle.

Fun fact: Mozilla bugathon.

Practice: Building a delta debugger.

Lesson 4: Tracking Origins

Theory: Cause-effect chain. Deduction. Dependencies. Slices.

Fun fact: Sherlock Holmes and Doctor Watson.

Practice: Improving the delta debugger.

Lesson 5: Reproducing Failures

Theory: Types of bugs (Bohr bug, Heisenbug, Mandelbug, Schrodinbug). Systematic reproduction process.

Fun fact: Mad laptop bug.

Practice: Building a statistic debugging tool.

Lesson 6: Learning from Mistakes

Theory: Bug database management. Classifying bugs. Bug maps. Learning from mistakes.

Fun fact: Programmer with the most buggy code.

Practice: Improving your tools and practicing on a real world bug database.

Outils

- [Valgrind \[http://valgrind.org/\]](http://valgrind.org/) (C, C++, Java, Perl, Python, ...)
- [QuickCheck \[https://en.wikipedia.org/wiki/QuickCheck\]](https://en.wikipedia.org/wiki/QuickCheck)
 - [JUnit-Quickcheck \[https://github.com/pholser/junit-quickcheck/\]](https://github.com/pholser/junit-quickcheck/) (Java)
- [GDB: The GNU Project Debugger \[https://www.gnu.org/software/gdb/\]](https://www.gnu.org/software/gdb/)
 - [Dix choses faisable qu'avec GDB \[http://devarea.developpez.com/dix-choses-faisables-avec-GDB/\]](http://devarea.developpez.com/dix-choses-faisables-avec-GDB/)

PHP

- [Xdebug \[http://xdebug.org/\]](http://xdebug.org/)

Profilage

- [GNU gprof \[http://sourceware.org/binutils/docs-2.18/gprof/index.html\]](http://sourceware.org/binutils/docs-2.18/gprof/index.html)
- [Kcachegrind \[http://kcachegrind.sourceforge.net/html/Home.html\]](http://kcachegrind.sourceforge.net/html/Home.html) est un front end pour Valgrind (version KDE). Il existe également une version Qt : Qcachegrind pour MS Windows et Mac OSX (à compiler depuis les sources obtenues sur le dépôt kcachegrind)

Java

PHP

- [XHProf \[https://github.com/phacility/xhprof\]](https://github.com/phacility/xhprof)
 - [XHProf.io \[http://xhprof.io/\]](http://xhprof.io/) (basé sur MySQL) (l'auteur déconseille son utilisation en production)
 - [Interface graphique pour XHProf \[https://github.com/perftools/XHGui\]](https://github.com/perftools/XHGui) (basé sur MongoDB)
- [Webgrind \[https://github.com/jokkedk/webgrind\]](https://github.com/jokkedk/webgrind) est un profileur basé sur Xdebug. Il reprend certaines fonctionnalités de kcachegrind

Analyse automatique du code

Java

- [Liste d'outils pour analyser automatiquement du code source \[http://java-source.net/open-source/code-analyzers\]](http://java-source.net/open-source/code-analyzers)

JavaScript

- [JSHint \[http://www.jshint.com/about/\]](http://www.jshint.com/about/)
 - [Pour utiliser JSHint en ligne \[http://www.jshint.com/\]](http://www.jshint.com/)

Analyse automatique de dépôt

- [scorecard: Security Scorecards - Security health metrics for Open Source \[https://github.com/ossf/scorecard\]](https://github.com/ossf/scorecard)

Optimisation du code

Généralités

- [Zen of Code Optimization de Michael Abrash \[http://www.amazon.com/Zen-Code-Optimization-Ultimate-Software/dp/1883577039\]](http://www.amazon.com/Zen-Code-Optimization-Ultimate-Software/dp/1883577039) (Algorithmique / C / Assembleur)

Java

- [Java Performance Tuning \[http://www.javaperformancetuning.com/\]](http://www.javaperformancetuning.com/) est un site entièrement dédié à l'optimisation des programmes écrits en Java.
 - La rubrique [tuning tips \[http://www.javaperformancetuning.com/tips/index.shtml\]](http://www.javaperformancetuning.com/tips/index.shtml) contient de nombreux trucs à connaître; ils sont classés par catégories.
 - La rubrique [ressources \[http://www.javaperformancetuning.com/resources.shtml\]](http://www.javaperformancetuning.com/resources.shtml) constitue une bonne base de départ. On y retrouve des sections concernant les livres, les outils de profilage, de tests et de benchmarks, les compilateurs natifs, les classes optimisées, les outils d'optimisation automatique, etc.
- Quelques bibliothèques optimisées
 - [GNU Trove \[http://trove4j.sourceforge.net/\]](http://trove4j.sourceforge.net/)
 - [fastutil: Fast & compact type-specific collections for Java \[http://fastutil.dsi.unimi.it/\]](http://fastutil.dsi.unimi.it/)
 - [Javolution \[http://www.javolution.org/\]](http://www.javolution.org/)

Latence (web)

Pour les applications web, il est possible d'améliorer la latence par l'utilisation d'un CDN (Content Delivery Network) qui héberge les bibliothèques les plus populaires au plus près de l'internaute.

- [Bibliothèques hébergées par Google \[https://developers.google.com/speed/libraries/\]](https://developers.google.com/speed/libraries/) (liste fixe)
- [CDNjs \[https://cdnjs.com/\]](https://cdnjs.com/) est hébergé par CloudFlare (liste fixe)
- [jsDelivr \[http://www.jsdelivr.com/\]](http://www.jsdelivr.com/) héberge des bibliothèques Javascript, jQuery, des fonts, des frameworks CSS,... (il est possible de soumettre d'autres fichiers)

Maintenance

- [7 techniques pour reprendre le contrôle de ton legacy code \[https://www.jesuisundev.com/7-techniques-pour-reprendre-le-contrôle-de-ton-legacy-code/\]](https://www.jesuisundev.com/7-techniques-pour-reprendre-le-contrôle-de-ton-legacy-code/)

Méthodes de numérotation de version

- [ZeroVer: 0-based Versioning \[https://0ver.org/\]](https://0ver.org/)
- [CalVer : Calendar Versioning \[https://calver.org/\]](https://calver.org/)
- [SemVer : Semantic Versioning \[https://semver.org/\]](https://semver.org/)

Mise en production

- [Why we don't use a staging environment \[https://squeaky.ai/blog/development/why-we-dont-use-a-staging-environment\]](https://squeaky.ai/blog/development/why-we-dont-use-a-staging-environment/)

Automatisation

- [wakeci: Task automation \[https://github.com/jsnjack/wakeci\]](https://github.com/jsnjack/wakeci)

Recherche de code source

Moteur de recherche

- [Open Hub \[https://www.openhub.net/\]](https://www.openhub.net/)

Forges (projets open-source)

- [http://sourceforge.net \[http://sourceforge.net\]](http://sourceforge.net)
- [http://code.google.com/intl/fr/ \[http://code.google.com/intl/fr/\]](http://code.google.com/intl/fr/)
- [https://github.com \[https://github.com\]](https://github.com)
- [http://www.tigris.org \[http://www.tigris.org\]](http://www.tigris.org)

Index de projets open-source

- [http://freecode.com \[http://freecode.com\]](http://freecode.com) permet une recherche par mots-clés/tags/langages et redirige vers le site web du projet.

Sécurité web

- [Nikto \[https://www.cirt.net/Nikto2\]](https://www.cirt.net/Nikto2) permet d'effectuer certains tests de sécurité sur un serveur web.

Divers

- [Inno Setup \[http://www.jrsoftware.org/isinfo.php\]](http://www.jrsoftware.org/isinfo.php) est un programme qui permet de faciliter l'installation de votre application sur les machines fonctionnant sous Windows.
- Site contenant de nombreuses informations sur Java [\[http://www.javakb.com/\]](http://www.javakb.com/) (forums de discussions, liste de projets, etc.)
- [Diagramme d'Ishikawa \[http://fr.wikipedia.org/wiki/Diagramme_d'Ishikawa\]](http://fr.wikipedia.org/wiki/Diagramme_d'Ishikawa)
- [Exemple de nombre de lignes de code : avion / voiture \[http://spectrum.ieee.org/transportation/systems/this-car-runs-on-code\]](http://spectrum.ieee.org/transportation/systems/this-car-runs-on-code)
- [NASA's Software Catalog \[https://software.nasa.gov/\]](https://software.nasa.gov/)
- [Une erreur de code sur StackOverflow perpétue un bug depuis plus de 10 ans \[https://korben.info/une-erreur-de-code-sur-stackoverflow-perpetue-un-bug-depuis-plus-de-10-ans.html\]](https://korben.info/une-erreur-de-code-sur-stackoverflow-perpetue-un-bug-depuis-plus-de-10-ans.html)
- [Getting started with contributing to open source \[https://stackoverflow.blog/2020/08/03/getting-started-with-contributing-to-open-source/\]](https://stackoverflow.blog/2020/08/03/getting-started-with-contributing-to-open-source/)

- De l'importance des process [<https://blog.i-run.si/posts/silife/2020-07-24-de-l-importance-des-process/>]

enseignement/gl/liens.txt · Dernière modification: 2022/04/05 10:58 de janiszek