

Remarques sur les cours master COMASIC

Organisation générale

Inscription et logement

Pour l'inscription vous avez sûrement déjà envoyé votre candidature. Vous recevrez la réponse d'ici fin mars (ne vous inquiétez pas ça sera positif). Après vous n'aurez sans doute plus de nouvelle jusqu'au mois de juin. Ils vont sûrement vous contacter à ce moment là pour les logements. S'ils font comme avec nous et vous proposent un logement sur le campus n'hésitez surtout pas, ils sont globalement très bien (colloc ou chambre de type celles de sup) et surtout les logements à Saclay c'est vraiment la galère en ce moment. N'hésitez pas aussi à contacter Eric Goubault pour qu'ils vous pistonnent pour les logements et globalement pour tout problème même mineur il est ultra sympa et toujours content d'aider. Pour l'inscription, ça ce fait dans le courant du mois de juillet, à peu près en même temps que Sup. Il faut juste remplir un formulaire et payer environ 270 € pour l'inscription. Gardez bien le reçu Thalès nous l'a demandé pour nous la rembourser fin décembre. Les cours ont commencé le 16 septembre avec avant une petite réunion (1h-1h30) le 4 pour expliquer comment se déroule l'année et présenter rapidement les différents cours. On a ensuite dû choisir nos cours (la plupart sont obligatoires mais on peut quelques fois choisir). Si vous êtes motivés, vous pouvez prendre tous les cours, par contre méfiez vous un peu de ça vous risquez d'avoir beaucoup de travail en fin de semestre, la plupart des cours se valident avec des projets qui doivent tous être rendu plus ou moins en même temps.

0.1 Organisation de l'année et quelques conseils

L'année est divisée en 2 semestres. Le premier c'est les cours (de mi septembre à fin février). Ce semestre est divisé en deux demi semestres (de mi septembre à mi novembre et de décembre à fin février) et vous suivez les cours en parallèle. Vous avez exactement le même emploi du temps toutes les semaines pour chaque demi-semestre (hors cours déplacé) et les cours ont lieu sur une demi journée (souvent 9h-12h ou 14h-17h). Le truc le plus pénible c'est que les cours ont tous pas lieu au même endroit, ça peut être le labo d'info de l'X (pour la plupart), les salles de PC de l'X, Télécom, l'ENSTA, le CEA ou la fac. Les deux derniers sont accessibles uniquement par bus mais c'est bien desservi. Pour presque tous les cours, vous avez besoin de vos ordinateurs (avec Linux ET Windows de préférence, Linux est indispensable et Windows est bien plus pratique pour certains logiciels). A quelques rares exception, chaque cours nécessite son propre logiciel (voir après). Après vous avez à disposition les PC des différentes écoles (sauf à la fac ils nous ont pas créé d'identifiant) mais déjà ça veut dire un identifiant/mdp par école et surtout les projets sont à faire chez soi, alors autant éviter de devoir sans arrêt passer d'un PC à l'autre. Ne soyez pas trop étonné en début de semestre vous n'aurez pas grand chose à faire les cours se suffisent souvent à eux même. Par contre essayez de commencer les projets le plus tôt possible pour pas vous retrouver déborder. Les partiels sont globalement assez faciles (bien plus qu'à Sup) et très peu nombreux.

Au deuxième semestre vous devez faire un stage à contenu scientifique fort (c'est ce qu'à dit M. Goubault en début d'année). Ca veut plus ou moins dire que ça doit être un stage de recherche. Allez le voir dans son bureau ou envoyez lui un mail pour lui dire quel type de stage vous voulez (lieu/sujet), il a plein de relation et ça évite de galérer à trouver. Ou autrement les profs proposent

aussi des stages mais c'est en France donc pas pratique si vous devez valider votre obligation internationale. Cette année, on a trouvé des stages à l'X, à Télécom, au Texas et à Munich. Sinon niveau administratif Nolwenn Marchaland et très sympa et efficace et globalement tous les profs sont très accessibles que ce soit par mail ou directement dans leur bureau.

Conseil plutôt important : si vous avez jamais touché à du Caml, regardez un peu le fonctionnement du langage c'est un prérequis à certains cours.

1 Les cours

Les intitulés des cours sont la plupart du temps longs et très vagues, c'est pour ça qu'on vous a fait une petite description de chaque cours avec nos avis (parfois très variables) et les sites des profs s'ils existent. De manière générale, les cours sont pas trop dur à valider, mais ils demandent tous une implication personnelle assez importante, surtout si, comme Charles et moi vous avez un background en info assez limité. Cette année on n'était que 4 dans le master (oui promis c'est pas une blague) avec quelques cours avec la fac où on était une quinzaine. Du coup les profs sont hyper accessibles (et tous vraiment très sympa) pendant les cours et les TP. Même si certains sont pas hyper doués pour rendre leur cours intéressant, ils sont tous très compétents et veulent vraiment nous apprendre des trucs pas juste cracher leur cours.

Architecture système et ingénierie dirigée par les modèles (2 UE obligatoires)

Modélisation objet pour les systèmes

Lieu : CEA (30 min en bus du campus)

Notation : un projet à faire en groupe avec une soutenance

Logiciel : papyrus (Linux ou Windows) <https://www.eclipse.org/papyrus>

Cours sur le MBSE, beaucoup mieux fait que celui de Sup (en tout cas j'espère pour vous qu'ils l'ont changé par ce qu'il était horriblement ennuyant cet électif). Une introduction à l'utilité du MBSE un peu vue et revue mais le prof se concentre vite sur les méthodes alors c'est plutôt intéressant. Pas de TP hormis un tutoriel sur l'utilisation de papyrus (logiciel de modélisation UML par très bien fait mais qui marche bien quand on s'y est fait). Un gros projet en groupe (au début de 2 mais on a fini par le faire à 4) assez compliqué de prime abord (nous on avait choisi les circuits quantiques) mais vraiment super intéressant quand on s'y plonge. Si vous vous sentez un peu perdu au début c'est normal, mais il ne faut absolument pas hésiter à harceler les profs durant les heures de cours consacrées au TP pour pouvoir en discuter avec eux. Il faut savoir que ce sont des projets de recherche, donc les profs n'ont pas vraiment de solution en tête du coup il peut y avoir des vrais discussions avec eux et pas juste des réponses vagues pour pas donner la solution.

Globalement cours très applicatifs (1/3 cours 2/3 projet) qui permet vraiment de réfléchir et de faire de la modélisation.

Sûreté de fonctionnement

Lieu : Télécom

Notation : un projet qu'on fait presque entièrement pendant le cours et un exam de 2h

Logiciel : prism (Linux uniquement) Site : <https://trobert.wp.imt.fr/fr/enseignements-2/comasic-sdf>

L'un des seuls cours dont le nom parle de lui-même. Il est surtout orienté sûreté de fonctionnement de logiciels et de programmes (ie comment augmenter la fiabilité d'une architecture logicielle ou

d'un code). Le sujet est très intéressant et le contenu du cours aussi. Le gros problème c'est que le prof est extrêmement ennuyant. Il parle sans s'arrêter pendant des heures (le cours est sur des plages horaires de 4h et ça dure souvent plutôt 4h30) et c'est souvent difficile de suivre pendant tout ce temps, surtout quand il n'y a pas de TP. Comme dirait Charles, le meilleur conseil qu'on puisse vous donner c'est de bien faire la sieste avant. Mais les TP et l'examen ne posent pas de difficulté. Pour le projet, le prof a même fini par nous donner toutes les réponses sans qu'on les lui demande. L'examen c'est principalement du cours avec quelques applications basiques et il y a beaucoup d'annales données sur son site donc vraiment pas de surprises.

Vérification et validation (2 UE obligatoires)

Validation inductive de programmes et de systèmes hybrides

Lieu : LIX

Notation : un TP noté, un projet et une soutenance, avec des exercices d'application directe du cours en DM

Logiciels : PAG (à installer avec virtual box, cherchez pas vous trouverez quasi rien sur internet) et UPPAAL

Site : <http://www.enseignement.polytechnique.fr/profs/informatique/Sylvie.Putot/Verif/index.html>

Le cours est séparé en deux parties. La première c'est la validation inductive de programme, qui parle principalement de l'interprétation abstraite et de l'analyse par intervalle (comment s'assurer qu'un code vérifie un certain nombre de contraintes comme par exemple $x \leq 10$). On a généralement 1h30 de cours et 1h30 de TP. Le TP utilise un langage totalement inconnu est très mal documenté, PAG. Il est noté mais la prof est ultra disponible pendant les cours pour nous aider. Il faut bien suivre le cours pour ne pas être perdu dans le TP et perdre du temps. C'est assez rébarbatif et la plateforme et le langage sont plutôt nuls. Après la méthode d'analyse par intervalle est intéressante et ça demande à réfléchir différemment de d'habitude quand on écrit du code.

La deuxième partie du cours est centré sur les systèmes hybrides (qui mélangent du logiciel et du matériel) et sur comment les modéliser et les analyser. Cette partie est plutôt pas mal, et sur certains points se rapproche du cours de sûreté de fonctionnement. D'ailleurs dans ce master beaucoup de cours sont reliés entre eux soyez pas surpris. On a le choix entre trois projets, l'un très pratique sur la modélisation et la vérification d'un pacemaker sur le logiciel UPPAAL (c'était mon choix j'ai trouvé ça vraiment cool) et deux autres un peu plus théoriques. Ne prenez pas celui sur les réseaux de neurones si vous ne voulez pas passer un temps infini à lire des articles et du code avant de pouvoir commencer le projet. La soutenance est assez soft, mais soyez surs de maîtriser votre projet pour ne pas avoir de questions de cours.

Sinon la prof est assez chiant, essayer de faire en sorte qu'elle ne fasse pas le cours en anglais, ça sera déjà mieux que nous. Ce cours m'a vraiment intéressé du coup c'était bien mieux que le cours de sûreté de fonctionnement, mais tout le monde n'est pas d'accord avec moi.

Vérification déductive de programmes

Lieux : fac

Notation : deux mini exams de 30min à la fin de chaque partie, 1 TP noté (à finir à la maison) et 1 exam Logiciel : frama-c (vous allez avoir des problèmes d'installation, mais ne vous embêtez pas à essayer d'installer la version la plus récente, la version du dépôt d'ubuntu suffit largement)

Site du cours : <http://why3.lri.fr/fiil-2019>

Cours vraiment hyper intéressant et bien fait, les deux profs sont super dynamiques et très sympas.

C'est de la vérification de programme, un peu comme on fait en prépa (terminaison, invariant de boucle) mais en beaucoup plus poussé. C'est assez déroutant et difficile au début, mais une fois qu'on y a passé un peu de temps, c'est rigolo. Le cours présente deux logiciels différents (avec deux profs différents), l'un en Caml et l'un en C, qui font globalement la même chose du coup si vous vous accrochez au début la suite est plutôt simple. Le seul point négatif c'est frama-c. Parfois le TP ne marche pas juste à cause d'un problème de version (par exemple, j'avais la version 2019 ça marchait pas mais pour les versions 2020 et 2018, celles du prof, ça marchait) donc soyez pas surpris. C'est le seul truc qui est un peu pénible de ce cours.

Plateforme matérielle et logicielle (2UE à choisir parmi 3)

Noyaux temps-réels

D'après Garion, cours redondant avec un des cours de 3A donc Charles et moi ne l'avons pas pris. Si vous voulez des infos, demandez à JR.

Systèmes réactifs et synchrones

Lieu : LIX

Notation : projet

Logiciel : langage Lustre

Site : <https://www.di.ens.fr/~pouzet/cours/comasic>

Cours plutôt intéressant qui traite d'un type de langage qu'on ne connaît pas du tout : les langages synchrones. Mais le prof est un prof de l'ENS Ulm, extrêmement fort et cultivé sur le sujet, du coup il part un peu souvent dans des délires un peu bizarre. Il est surtout assez mou (un peu comme Pastor) et ses cours sont souvent barbant et répétitifs. C'est l'un des seuls cours où il vaut bien mieux lire les slides et les pdf tranquille chez soit plutôt que d'essayer de l'écouter.

Le projet est très difficile et exigeant. Il faut écrire un compilateur en caml qui permet de passer d'un langage synchrone à un langage impératif (typiquement du C). Prenez y vous très en avance et n'attendez pas que le cours vous aide plus que ça. On aurait pu commencer le projet dès la fin du deuxième cours mais comme on était perdu, on a cru que les cours suivants nous aideraient, ce qui n'était absolument pas le cas. Il faut passer beaucoup de temps pour comprendre comment ça marche, surtout au niveau de l'organisation des fichiers fournis par le prof. Après une fois que tu commences à comprendre le principe, j'ai trouvé ça plutôt intéressant alors n'hésitez surtout pas à me poser des questions si vous n'y arrivez pas.

Systèmes répartis et autonomiques

Lieu : Télécom

Notation : un QCM et un projet (avec mini soutenance et sans rapport)

Logiciel : extension d'Eclipse hyper facile à installer si vous l'avez déjà fait pour le cours de Java.

Site : <https://perso.telecom-paristech.fr/diacones/comasic>

La prof est hyper (même un peu trop parfois) dynamique et le cours globalement intéressant, surtout quand on passe au projet. L'idée générale du cours c'est de rendre des applications java dynamiques (par exemple pouvoir créer/détruire des instances de classe pendant la runtime). La première partie du cours est moyennement intéressante et les TP inutiles (surtout du copier-coller de fragment de codes) mais il est très important de bien comprendre ce qu'on fait pour la suite. La suite du cours, le projet iCasa, est très bien guidée et ludique c'est plutôt cool à faire. Les cours sont tout le temps coupés en deux entre cours et TP ce qui permet de pas trop s'ennuyer.

Gros point noir pour moi : le QCM. C'est juste du recrachage par coeur de la première partie du cours (question du style quelle est le nom de la méthode qui permet de faire tel truc). Mais bon il suffit de bien refaire les TP et relire les cours et la validation est assez simple. La mini soutenance consiste juste à montrer à la prof ce qu'on a fait pour le projet, elle est très pointilleuse sur à peu près tout alors préparez le un minimum mais pas la peine de se mettre trop la pression.

Traitement du signal et contrôle (2UE obligatoires)

Signaux et systèmes

Cours dont on a été dispensé car très proche du cours d'autom de 2A.

Modélisation et commande par représentation d'état

Lieu : LIX

Notation : projet à rendre à la fin (à peu près 5 cours entiers y sont consacrés donc vous avez le temps)

Logiciel : matlab

Au début c'est assez redondant avec les cours de Bérard mais on aborde vite des trucs nouveaux essentiels pour les projets. Le prof a pas du tout la même vision de l'autom que Bérard (beaucoup plus mathématique et moins physique) donc les deux cours sont très complémentaires et les TP/-projet sont sur matlab mais sans simulink. Le prof est plutôt un bon pédagogue même si souvent on a un peu l'impression de le déranger (il est au fond en train de travailler pendant les TP). Le projet est vraiment cool, il y a des sujets pour tous les goûts (ça va du contrôle d'un amortisseur de voiture à l'étude du rôle du dosage des médicaments dans le traitement de l'hépatite B). Ne soyez pas étonnés si les sujets sont parfois un peu bizarre et pas très en accord avec le cours, le principal c'est de se montrer actif et de ne pas hésiter à recourir au prof pour discuter avec lui des modèles.

Systèmes distribués (1UE au choix)

Algorithmique répartie

Lieu : fac

Notation : un exposé et un exam

L'un des deux seuls cours de l'année où on ne touche pas à un ordinateur. Mais il est très intéressant et pas trop mal fait dans l'ensemble. Il permet de traiter un pan de l'info qu'on connaît pas du tout (comment faire fonctionner ensemble plusieurs sites qui utilisent tous le même algorithme). L'exposé est à faire à deux, il faut choisir entre plusieurs articles scientifiques poussés (qui consistent souvent en la description d'un ou plusieurs algorithmes) et le vulgariser pour que tous les élèves le comprennent. Les seuls conseils qu'on peut vous donner c'est de ne parler que de ce que vous maîtrisez (il ne faut pas traiter l'article en entier) au risque de vous faire bombarder de question sur le mini point que vous n'avez pas compris et de se forcer à suivre 1 autre exposé parce qu'il y a des questions assez précises à l'exam.

C'est quasi le seul cours où l'exam est un peu difficile. Prenez bien des notes pendant les cours, il n'y a ni slides ni poly et l'exam consiste plus ou moins à refaire les exemples du cours avec quelques variantes. Ce cours donne aussi une bonne idée de ce que peut être la recherche en informatique théorique, les profs sont très disponibles pour en parler à la fin des cours.

1.0.1 Réseaux de capteurs

Lieu : Télécom

Notation : un projet à rendre (à faire presque exclusivement chez soi)

Logiciels : Fit-Iot Lab et Cooja mais n'essayez pas tout seul vous n'allez rien comprendre

Cours en théorie intéressant où on en apprend pas mal sur les protocoles d'échanges entre objets connectés et sur les communications cellulaires. A part ça j'ai trouvé le reste extrêmement laborieux et pour être honnête chiant. Les TP ne sont que des manipulations sur la console où l'on ne comprend pas grand chose de ce qu'on fait. C'est uniquement lors du dernier cours qu'on utilise Cooja qui permet de plus se concentrer sur ce qui se passe et non sur des bugs mineurs essentiellement dus à la console. Après le contenu est plutôt intéressant. A choisir entre les deux cours, je préfère largement le premier mais si l'algorithmique pure ne vous intéresse pas, ou si l'IoT vous intéresse beaucoup les deux cours peuvent être bien.

1.1 Analyse et résolution d'équations différentielles algébriques

Lieu : ENSTA

Notation : un projet que l'on fait essentiellement chez soi

Logiciels : matlab et DynIbex

Site : <https://perso.ensta-paris.fr/chapoutot/integration/>

Cours sur la résolution d'équations différentielles qui consiste essentiellement à lister des méthodes d'intégrations dont certaines ont été vues en prepa (méthode d'Euler, Runge-Kutta, ...). Le prof ne fait que lire ces diapos qui sont un véritable cours. Encore une fois, essayez de faire en sorte qu'il ne fasse pas le cours en anglais, au risque de le voir galérer et demander ses mots. Heureusement, les cours ne durent qu'1h30 (ou 2h s'il prend trop de retard avec son anglais) et le TP qui suit est plutôt cool et facile. Je vous conseille d'utiliser matlab pour les faire parce que ça vous entraîne pour le projet. Le projet était plutôt intéressant, mais le sujet peu clair et mal présenté. Posez bien des questions lors de la séance de présentation, ça ne peut que vous aider.

1.2 Modélisation et analyse des risques de sécurité dans les systèmes complexes

Lieu : salles de TP de l'X mais probablement à Télécom pour vous

Notation : un exam qui ressemble plus à une blague quand t'as fait les annales.

Cours d'introduction à la cyber-sécurité et à la cryptanalyse. Le prof est un vrai passionné et donne énormément d'exemples pratiques et tous très intéressants. On y aborde plusieurs thématiques liées à la sécurité informatique, toujours de façon superficielle mais suffisamment pour nous donner envie de creuser si le sujet nous parle. Les notions abordées restent basiques, ce qui rend les TP globalement simples. Quand à l'examen, le prof a littéralement repris, de manière consciente, des exercices entiers des annales qu'il nous avait donné la semaine d'avant. Méfiez vous quand même, pas sûr qu'il fasse ça tous les ans.

1.3 Spécialisation (1 UE au choix)

Je préviens direct, cette UE c'est un peu choisir entre la peste et le choléra.

1.3.1 Modélisation métier : système de transport

Lieu : LIX

Notation : un QCM et un projet qu'on fait entièrement en cours.

Logiciel : arKitect (sur Windows uniquement) Ce cours est séparé en deux parties totalement disjointes. La première est plus ou moins un cours de sûreté de fonctionnement, mais plus général que l'autre puisqu'on y aborde largement la partie matérielle. Le prof nous prend parfois un peu pour des gamins mais ça fait du bien parfois de ne pas avoir trop réfléchi tout en apprenant des trucs. Le QCM reprend bien le cours est assez simple.

La deuxième partie est pour le coup un peu bizarre. Ça consiste à utiliser un logiciel industriel (le prof est le PDG de la start up qui l'a conçu) qui sert à créer une représentation multi-échelle d'un système (vision fonctionnelle, vision matérielle, ...). Le projet consiste à modéliser un système de verrouillage des portes d'une voiture. Encore une fois, vous allez partir de pas grand chose alors n'hésitez pas à monopoliser le prof même s'il raconte souvent n'importe quoi. Et petit conseil, s'il vous demande si vous voulez créer votre propre méta-modèle, dites non même si vous savez pas ce que c'est, nous on l'a su qu'après et c'est ultra chiant.

1.3.2 Ingénierie des exigences

Lieu : LIX

Notation : un QCM et un projet qu'on n'a jamais pu rendre (voir après)

Logiciel : objectiver

Le cours est aussi séparé en deux parties qui sont censées être une partie théorique et une partie pratique.

La partie théorique est juste ennuyante à un point où l'on regrette les amphis de Massenot. La prof, une retraitée de chez Thalès, ne s'arrête jamais de parler pendant 3h (voire 3h30 parce qu'elle continue à l'infini), elle enchaîne les digressions (qui le plus souvent consistent à raconter sa vie) pour un contenu finalement assez faible voire franchement évident parfois. C'est un peu un cours de gestion de projet version exigence. Le QCM n'est pas très difficile avec toutefois quelques subtilités. La deuxième partie du cours a été pour nous un calvaire, mais je pense que ça devrait changer l'an prochain. La prof était injoignable, on a mis un mois à organiser un cours et après elle n'a plus jamais répondu jusqu'à début mars où elle nous a donné nos notes qui étaient en fait celles du QCM. En théorie, on aurait dû utiliser objectiver, un logiciel qui permet de faire du traitement des exigences mais on peut pas vous en dire beaucoup plus vu ce qu'on a réussi à faire.

Pour résumer, le premier cours est quand même largement mieux.

1.4 Formation (1 UE au choix)

1.4.1 Formation à la vie en entreprise

Lieu : fac

Notation : projet et soutenance

Le titre est un piège, et devrait être création et gestion d'entreprise. Le principe est le même que le projet ski des vacances d'hiver de 1A, en plus détaillé et plus complexe. Priez pour avoir une équipe motivée avec des gens sympas, c'est le seul truc qui permette de rendre le projet divertissant. Et petit conseil pour la strat' : visez le bas de gamme en location de masse ou le très haut de gamme avec un parc immobilier faible mais dont vous êtes proprio (requiert un emprunt initial). C'est les deux façons de dominer le marché qui on fonctionné cette année.

1.4.2 Formation à la recherche

Lieu : LIX

Notation : projet et soutenance

Logiciel : modelica et matlab

Le titre est une nouvelle fois un piège. Il s'agit d'un cours qui explique comment résoudre informatiquement des DAE qui est un bon complément au cours sur les ODE (voir après). Le prof est un très jeune chercheur très dynamique. Mais il ne faut pas avoir peur de refaire des maths comme en prépa, avec de l'algèbre et des polynômes. Le projet est un vrai projet de recherche. On est totalement libre dans le choix du sujet, la seule contrainte est que cela implique une équation différentielle qu'il faut analyser. Attention toutefois au choix du sujet, assurer vous avant de le faire valider par le prof qu'il ne soit pas trop dur, sinon vous allez vraiment galérer. J'avais choisi un sujet que le prof a a priori trouvé un peu trop simple au vu de la note finale alors faites attention à ça aussi. La soutenance se passe globalement comme les autres, le prof est très pointilleux sur à peu près tout et passe son temps à vous couper pour poser des questions.

Pour choisir entre les deux projets, il faut savoir que le premier a lieu sur tout une semaine (où vous faites que ça) et que cette année il a eu lieu à la toute fin du semestre quand on avait tous les projets à rendre. Ce que je peux vous conseiller si vous hésitez c'est de prendre les deux, de voir ensuite quand ils tombent dans l'année et d'envoyer un mail à Nolwenn Marchaland pour dire que vous voulez pas suivre l'un des deux si besoin (c'est ce que j'ai fait et je n'ai eu vraiment aucun problème).

1.5 *Projet robotique*

Lieu : LIX

Notation : un projet à rendre (qui constitue l'ensemble de ce cours)

Logiciel : matlab (avec une partie en C aussi).

Ce projet est essentiellement la continuité du cours d'autom et exactement le même principe que le projet de modélisation. C'est pas hyper intéressant au début parce qu'on refait plus ou moins exactement la même chose (et même deux fois une fois en matlab et une fois en C). La suite consiste à couper tout ça pour faire un client/serveur comme dans le cours de Java. C'est assez laborieux pour un résultat final pas hyper intéressant. La seule nouveauté c'est l'utilisation des buffers dans la dernière partie. Je dirais que ce cours est un bon résumé de tout ce qu'on a pu faire en autom cette année.

A part ça si vous avez des questions n'hésitez surtout pas. Un dernier conseil : si ce n'est pas déjà fait, faites vous un template en latex pour les rapports de projets vous allez devoir en écrire un bon paquet.