

# Aspects économiques Vaut-il mieux investir dans le test et éliminer tous les bogues ou bien accepter leur présence et les éliminer au fur et à mesure? L'idéal qu'il n'y ait pas de bogues en adoptant une démarche de qualité très stricte Coût de l'anomalie Chaque anomalie peut mettre en péril l'activité de l'entreprise, elle est donc proportionnelle aux dommages causés Le développeur peut corriger immédiatement l'anomalie et diffuser la correction L'application ne peut plus être utilisée car les dysfonctionnements sont considérables

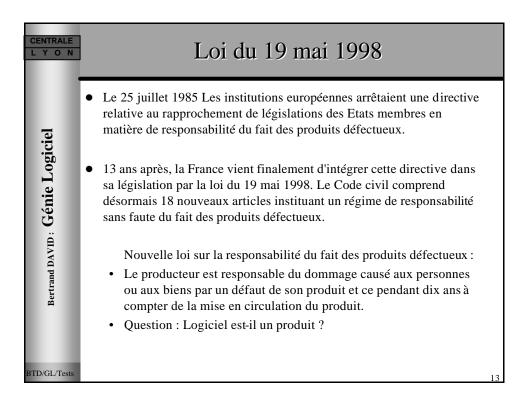
CENTRALE L Y O N	Le coût des tests
el	<ul><li>33% en développement</li><li>67 % en maintenance</li></ul>
Bertrand DAVID: <b>Génie Logiciel</b>	<ul> <li>Plus on teste tôt, plus on a de chances de trouver à temps les erreurs importantes</li> <li>Le test doit être intégré dans toutes les phases du cycle de développement</li> </ul>
BTD/GL/Tests	<ul> <li>Coûts des bogues détectés pendant le développement</li> <li>Le coût du test est de l'ordre de 20% du coût total du développeur, du chef de projet et des outils spécialisés</li> <li>Dans les phases d'intégration les bogues détectés coûtent 10 fois plus (car plus complexe à trouver et à corriger)</li> <li>Dans la phase de recette on passe à 100 fois plus</li> </ul>

# Coût des bogues détectés après le déploiement Pour l'utilisateur : coût direct et indirect de l'arrêt d'une partie de l'activité Pour l'éditeur ou le prestataire : coût de correction, frais d'un nouveau déploiement, paiement de pénalités Il est donc vital d'investir dans les tests et dans des méthodes de développement améliorant la qualité.

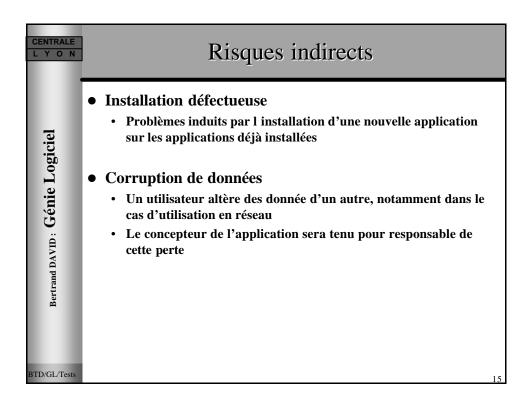
### Conséquences juridiques • Relations éditeur-utilisateur • Cas du logiciel sous plastique : Bertrand DAVID: Génie Logiciel • l'éditeur ne peut pas garantir que son logiciel soit exempt de t oute anomalie et ne peut prédire le contexte d'usage - il se décharge de toute responsabilité directe ou indirecte liée à l'usage du logiciel par la formule : • Garantie et responsabilités : « LLL » et la documentation qui l'accompagne (consignes d'emploi comprises) sont fournis dans l'état ou ils se trouvent et sans aucune garantie. En cas de support défectueux, un autre exemplaire sera délivré par « CCC » sur demande. « CCC » décline toute responsabilité découlant d'un dommage direct ou indirect en relation avec l'utilisation ou l'impossibilité d'utilisation de « LLL », y compris les dommages entraînés par la perte de bénéfices, l'interruption des activités, la perte d'informations et autres, et ce même si « CCC » a été informé de la survenance ou de l'éventualité de tels dommages. Comme l'acheteur n'achète pas le logiciel, mais une licence d'ut ilisation, l'éditeur retire sa responsabilité sur son exploitation. => L'acheteur devient responsable des dommages liés par l'usage d'un produit dont il n'a pas la maîtrise BTD/GL/Tests

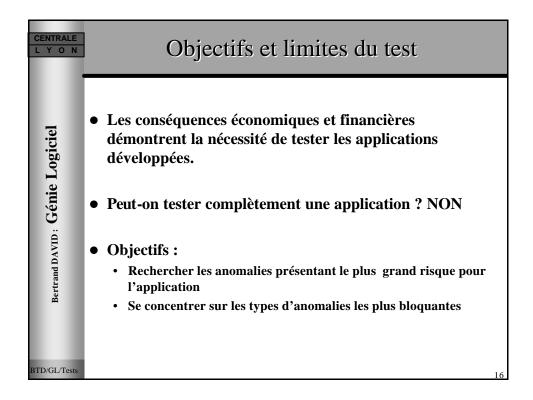
### Relations client-SSII (1/2) • Les contrats des relations entre SSII et donneurs d'ordres contiennent une rubrique ayant la forme suivante : Jeux d'essais en vue de recette Bertrand DAVID: Génie Logiciel • « FFF » devra fournir des jeux d'essais permettant de s'assurer que le programme mis en œuvre par elle contient bien aux besoins de « CCC » • Les jeux d'essais seront sous la responsabilité de « CCC » même si le « FFF » est amené à participer à leur mise au point. • Réception du programme • Le logiciel réalisé par « FFF » sera considéré comme conforme après mise en place sur les systèmes de « CCC » et contrôle de la validité à la suite d'un traitement effectué sur les jeux d'essais • « FFF » communiquera les résultats des essais à « CCC » qui devra s'assurer qu'ils sont conformes aux prestations demandées. • Si les jeux d'essais n'ont pas été remis dans les conditions prévues, ou si « CCC » n'a pas fait de réclamation écrite dans les dix jours qui suivent l'envoie des résultats, les programmes fournis seront considérés comme satisf aisants et acceptés en l'état par « CCC ». • La réception des résultats, sans réclamation ou contestation, vaudra recette définitive et la responsabilité ultérieur de « FFF » se trouvera dégagée en BTD/GL/Tests 1 'absence de réserves justifiées et formulées par par « CCC »

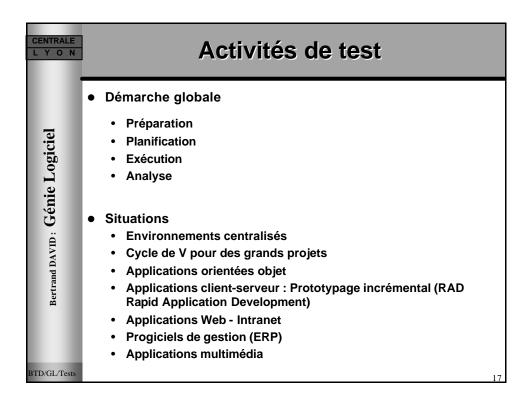
CENTRALE L Y O N	Relations client-SSII (2/2)
Bertrand DAVID: Génie Logiciel	<ul> <li>Registre des anomalies</li> <li>Il sera tenu un registre sur lequel « CCC » indiquera toutes les anomalies ou incidents remarqués concernant les logiciels en utilisant le classement qu'elle a défini dans le document</li> <li>Les représentants de « FFF » indiqueront les solutions adoptées pour y remédier, les résultats obtenus et les recommandations nécessaires</li> <li>Garantie</li> <li>Les logiciels standard éventuellement fournis sont garantis pendant une période de XXX mois à compter de la réception définitive de l ensemble du système.</li> <li>Cette garantie couvre toute anomalie (bogue) constatée, la remise gratuite de toute nouvelle version de logiciels (release) et la mise à jour (update) gratuite en cas de modification des règles comptables, fiscales ou juridiques mettant en cause la pertinence du logiciel livré.</li> <li>Pendant la durée de la garantie, la maintenance sera assurée par téléphone ou, si nécessaire, sur site dans le cas où »CCC » rencontrerait un problème particulier et ce sans aucun frais pour « CCC ».</li> <li>Les prestations de maintenance sont prévues à titre gratuit pendant la période de garantie.</li> </ul>
BTD/GL/Tests	Il est souhaitable d'inclure dans le CC l'utilisation d'outils

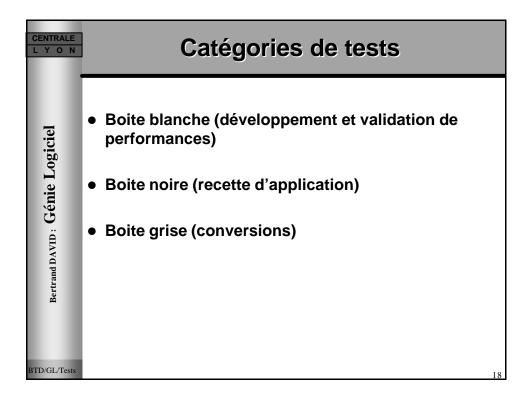


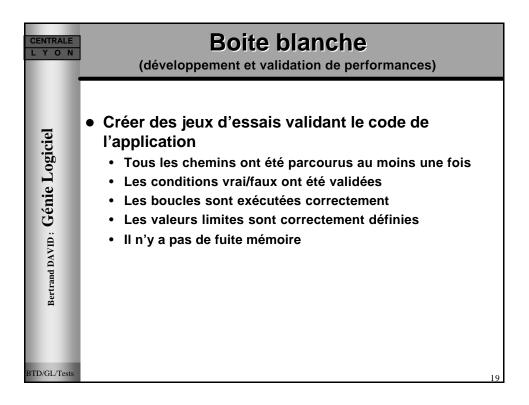
CENTRALE L Y O N	Le cas des logiciels
Bertrand DAVID: Génie Logiciel	<ul> <li>Sécurité et défaut</li> <li>Le bogue d'un logiciel à l'origine d'une perte de données constituera-t-il un défaut causant un dommage aux biens parce qu'il n'offre pas la sécurité à laquelle on peut légitimement s'attendre?</li> <li>Cas d'un logiciel intégré dans un appareil : « composant » du produit</li> <li>La notion de bogue fait référence à un dysfonctionnement par rapport à des exigences définies au préalable plutôt qu'à la notion de sécurité.</li> <li>Il faut donc préciser les conditions d'usage offrant le maximum de sécurité pour que cet article de la loi ne puisse s'appliquer.</li> </ul>

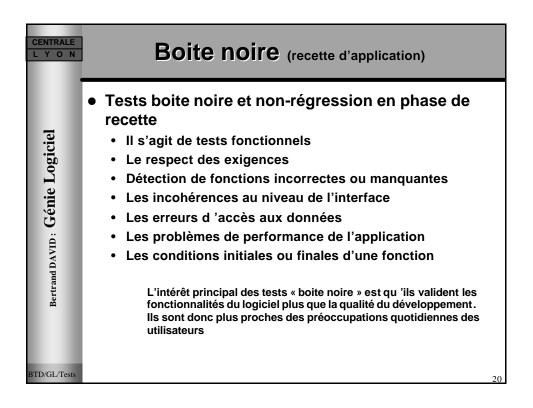




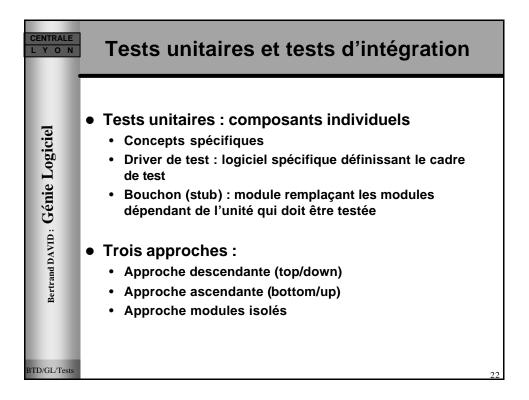




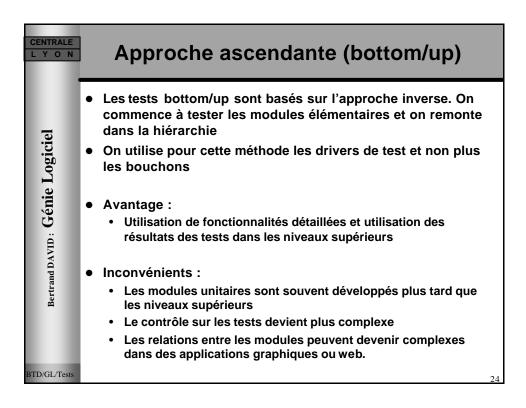








### Approche descendante (top/down) L'approche top/down consiste à tester chaque unité indépendamment de chaque unité appelante, en partant du haut de la hiérarchie. Chaque module appelé est remplacé Bertrand DAVID: Génie Logiciel par un bouchon. Avantages: • On effectue simultanément des tests unitaires et des tests d'intégration On valide dès le début les fonctions les plus importantes de l'application, car ces tests sont basés sur les besoins fonctionnels La structure de ces tests pourra être réutilisée dans les tests de non-régression Inconvénient · Utilisation de bouchons, dont la mise en œuvre peut être complexe. BTD/GL/Tests



# Approche modules isolés Le module est testé indépendamment des autres et on combine les deux approches top/down et bottom/up. On remplace ainsi chaque module appelant par un driver de test et chaque module appelé par un stub. Il est ainsi facile de tester chaque module, quels que soient la complexité de l'application et son impact sur les autres modules. En revanche, on perd la dynamique de l'application, et les tests d'intégration devront être conçus et réalisés indépendamment des tests unitaires

CENTRALE L Y O N	Tests de performance
Bertrand DAVID: Génie Logiciel	<ul> <li>Temps de réponse dans les conditions normales</li> <li>Temps de réponse dans les conditions extrêmes</li> <li>Volume</li> <li>Conditions extrêmes d'exploitation (charge utilisateur, charge réseau, manque de périphériques,)</li> </ul>
Bertrand DAVID: (	<ul> <li>Outils :         <ul> <li>Simuler les utilisateurs virtuels</li> <li>Enregistrement des scripts de simulation</li> <li>Exécution des scénarios de simulation des utilisateurs</li> </ul> </li> </ul>

