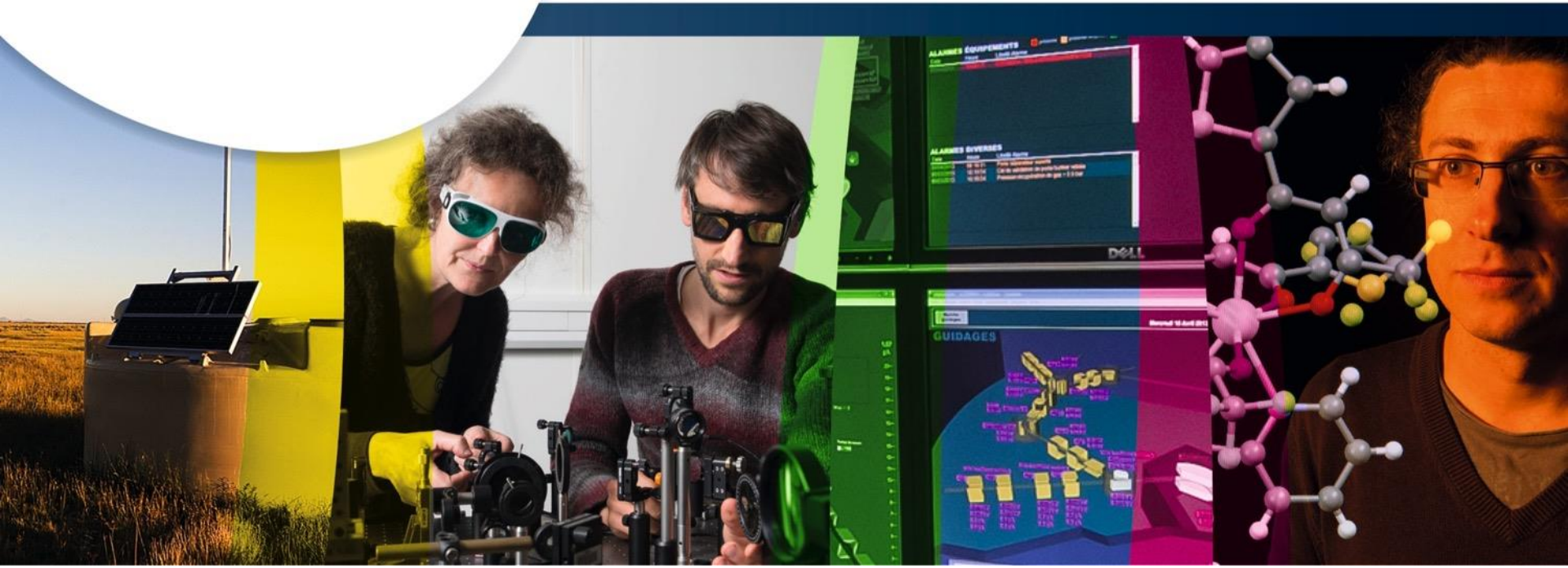
www.cnrs.fr

Organiser et mettre en œuvre la sécurité dans les applications Web
Méthodologies



Pour une approche pragmatique

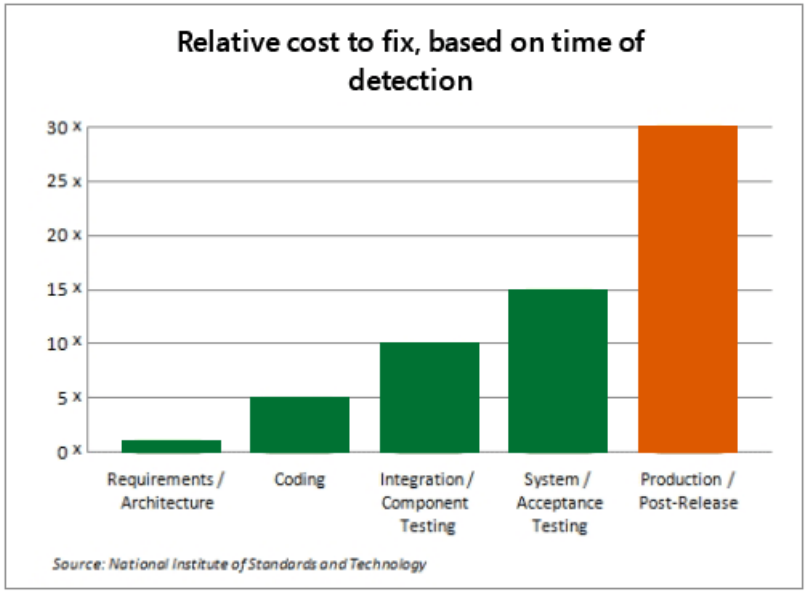
www.cnrs.fr

- Mission :
  - **Sensibiliser** à la sécurité des acteurs du développement
    - développeur ou responsable de développement
- Pourquoi :
  - Au cœur de la genèse des applications web
    - Parfois les seuls acteurs « informatiques »
  - La mise en œuvre applicable dès maintenant
- Objectifs :
  - Enraciner un mode de pensée
  - Donner les moyens de le faire grandir et le rendre effectif
  - Propager la culture sécurité
  - Rester pragmatiques

Security by design

www.cnrs.fr

- La sécurité, ce n'est pas *qu'un* combat d'admin système & réseau
- Comment justifier auprès de votre responsable ?
  - Coût d'un défaut par phase de détection
    - **1** en début de projet
    - **5** fois au codage
    - **10** fois en intégration/test
    - **15** fois en recette
    - **30** fois en production
- Économiser 1 jh de réflexion au début,
- Payer 30 jh de correction à la fin.



www.cnrs.fr

Introduction

Prendre la mesure des défis

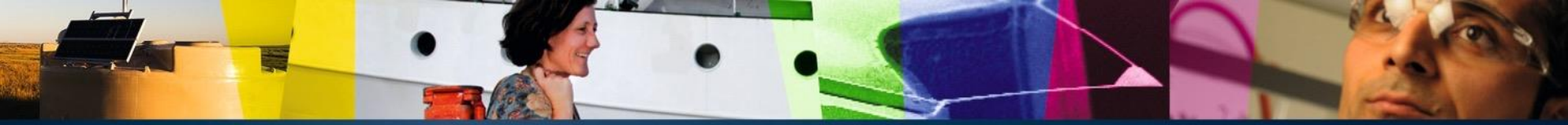
Les défis: l'insouciance et l'indifférence

www.cnrs.fr

- Au début est **l'insouciance**:
  - *« absence de souci, d'inquiétude, de tracas »*
  - Par simple ignorance ou focalisation
- Puis vient **l'indifférence**:
  - *« qui ne se sent pas concerné, touché par quelque chose »*
  - Biais d'optimisme : cela n'arrive qu'aux autres
  - Idées fausses : nous sommes *inintéressants, invisibles, infaillibles...*
- Manque de culture sécurité
  - Chefs de projet et responsables peu sensibilisés
  - Peu ou pas enseignée
  - Dissonance…

Valeurs conservatrices,
vocabulaire militaire

Valeurs de partage,
idées progressistes



Les défis:
le vertige de l'asymétrie

www.cnrs.fr

- Asymétrie de surface
  - Se protéger de milliers de failles possibles,
  - L'attaquant n'a besoin d'en trouver qu'une seule
- Asymétrie d'attention
  - Des failles détectées quotidiennement
  - Être sous la menace en permanence
  - Obligation d'alerte
  - ➔épée de Damoclès
- Conséquences
  - Déni et découragement

Apprivoiser le risque

www.cnrs.fr

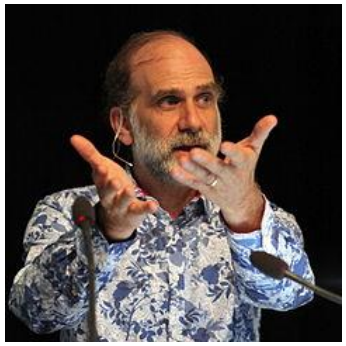
- **Accepter** la situation
  - **Il y a** des vulnérabilités dans vos applications
    - Comme il y a des bugs
  - La stabilité de la sécurité est éphémère
  - Cela n'a pas de fin
- **Apprivoiser** le risque
  - **Objectivement** pour vaincre l'indifférence et l'insouciance
  - **Progressivement** pour ne pas avoir le vertige
    - En fonction des moyens
    - En fonction de la maturité
- Ne plus penser **état, solution** mais **processus**

Penser processus

www.cnrs.fr

“Security is a process, not a product”

Bruce Schneier



… Products provide some protection, but the only way to effectively do business in an insecure world is to put processes in place that recognize the inherent insecurity in the products.
The trick is to reduce your risk of exposure regardless of the products or patches.
https://www.schneier.com/crypto-gram/archives/2000/0515.html#1

Plan

www.cnrs.fr

- Les chemins que nous allons prendre
  - Progression
  - Méthodologies
  - Analyse de risque et modélisation des menaces
  - Pratiques de développement
  - Principes
- **Objectif** : *que vo*us construisiez votre feuille de route ✓

Progression, maturité & expertise

www.cnrs.fr

Le jeune marche plus vite que l'ancien, mais l'ancien connait le chemin

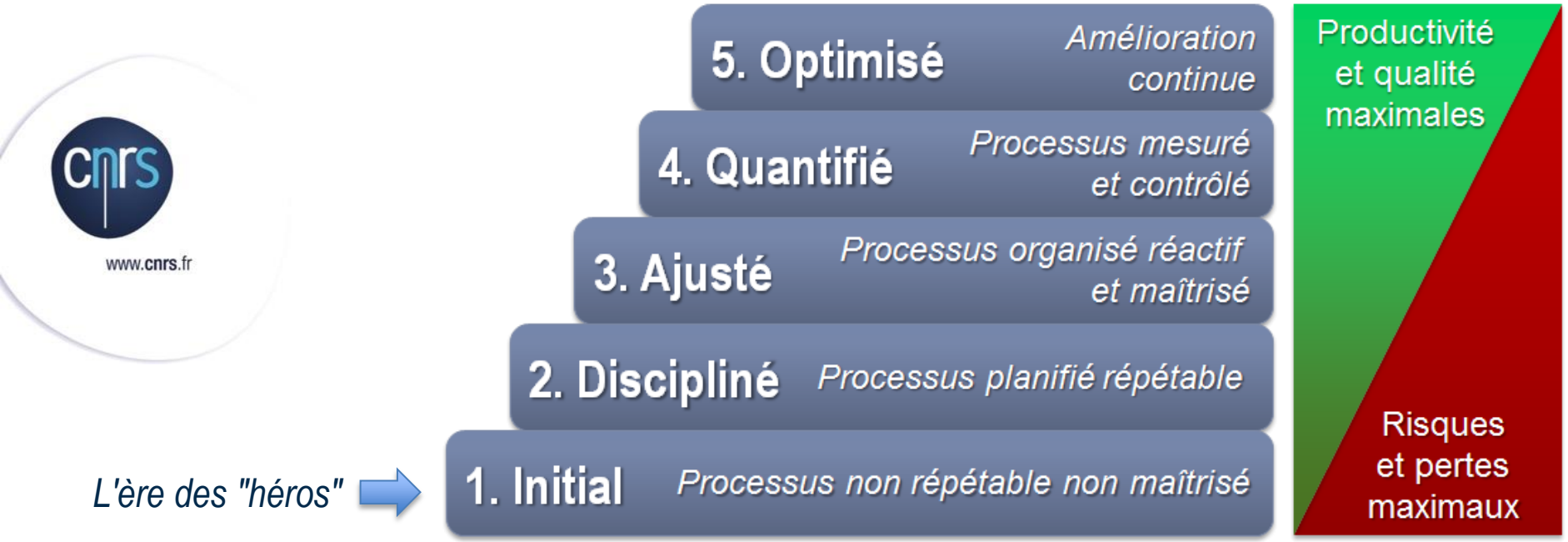
Maturité et expertise

www.cnrs.fr

- **Progresser**
  - Savoir où vous êtes
  - vers quoi vous devez aller.
- Passer de la ***survivance*** à ***l'efficacité***
- **Maturité** ➔ organisation
  - Illustration avec CMMI
- **Compétence** ➔individu ou équipe
  - Modèle de Dreyfuss & Dreyfuss

Maturité d'une organisation
Exemple CMMI

- CMMI Capacity Maturity Model Integration



L'ère des "héros"

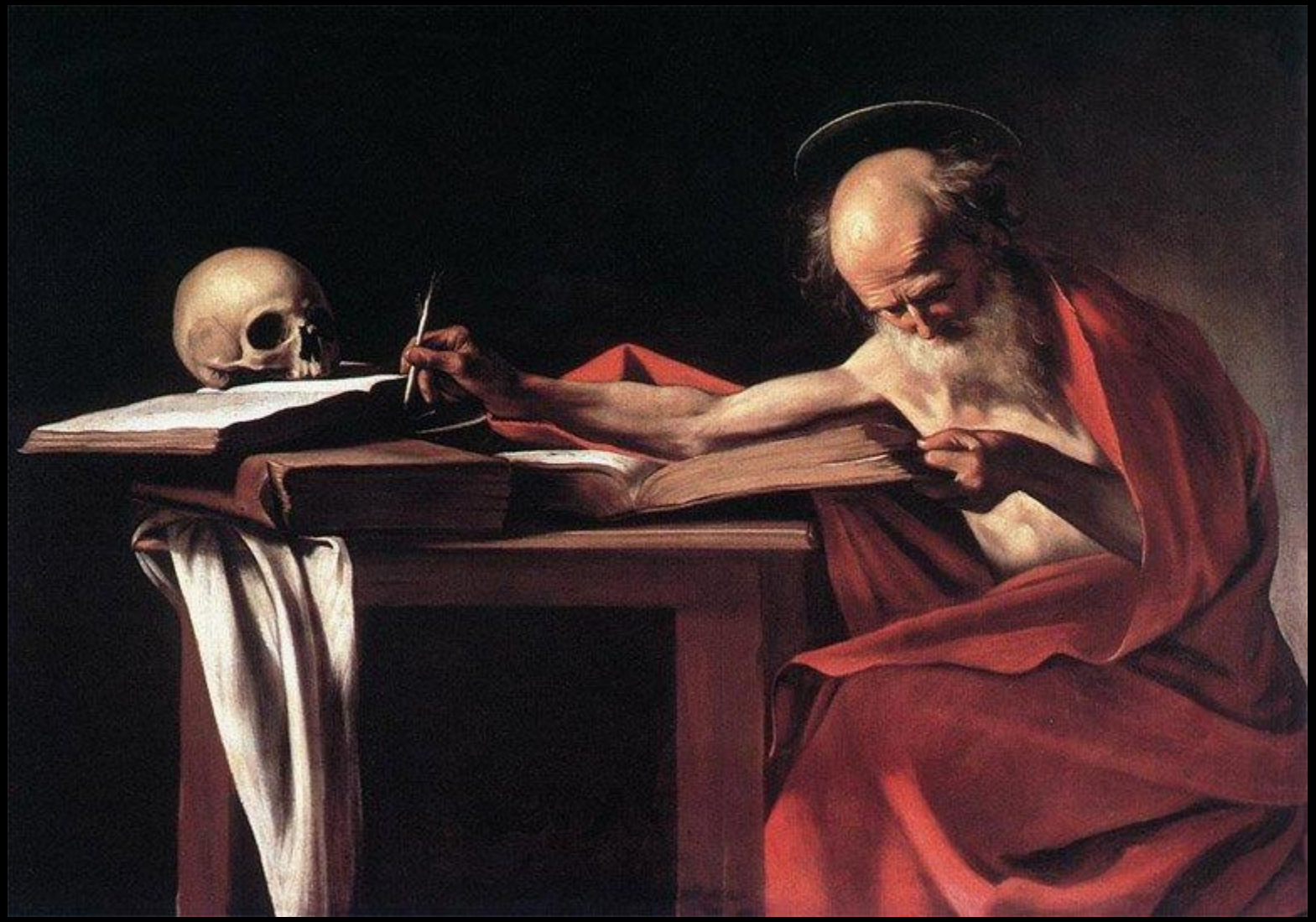
- Très bon article sur https://aresu.dsi.cnrs.fr/spip.php?article181 ☺

Montée en compétence
Modèle de Dreyfuss et Dreyfuss

www.cnrs.fr

- Sujet *délicat*
  - **Nécessaire** pour estimer ses capacités et ses faiblesses
  - Niveau sécurisation ~ niveau de compétence
- Modélisation de la montée en compétence par Dreyfuss & Dreyfuss
  - Mis en valeur dans « *Pragmactic Programmer* »
    - Mis au point pour les pilotes de chasse
    - Repris dans les monde infirmier (Patricia Brenner)
- 5 niveaux
  - Novice → débutant confirmé → efficace → compétent → expert

Choisissez un domaine de compétence de votre vie professionnelle…



Niveaux de compétence (1/2)

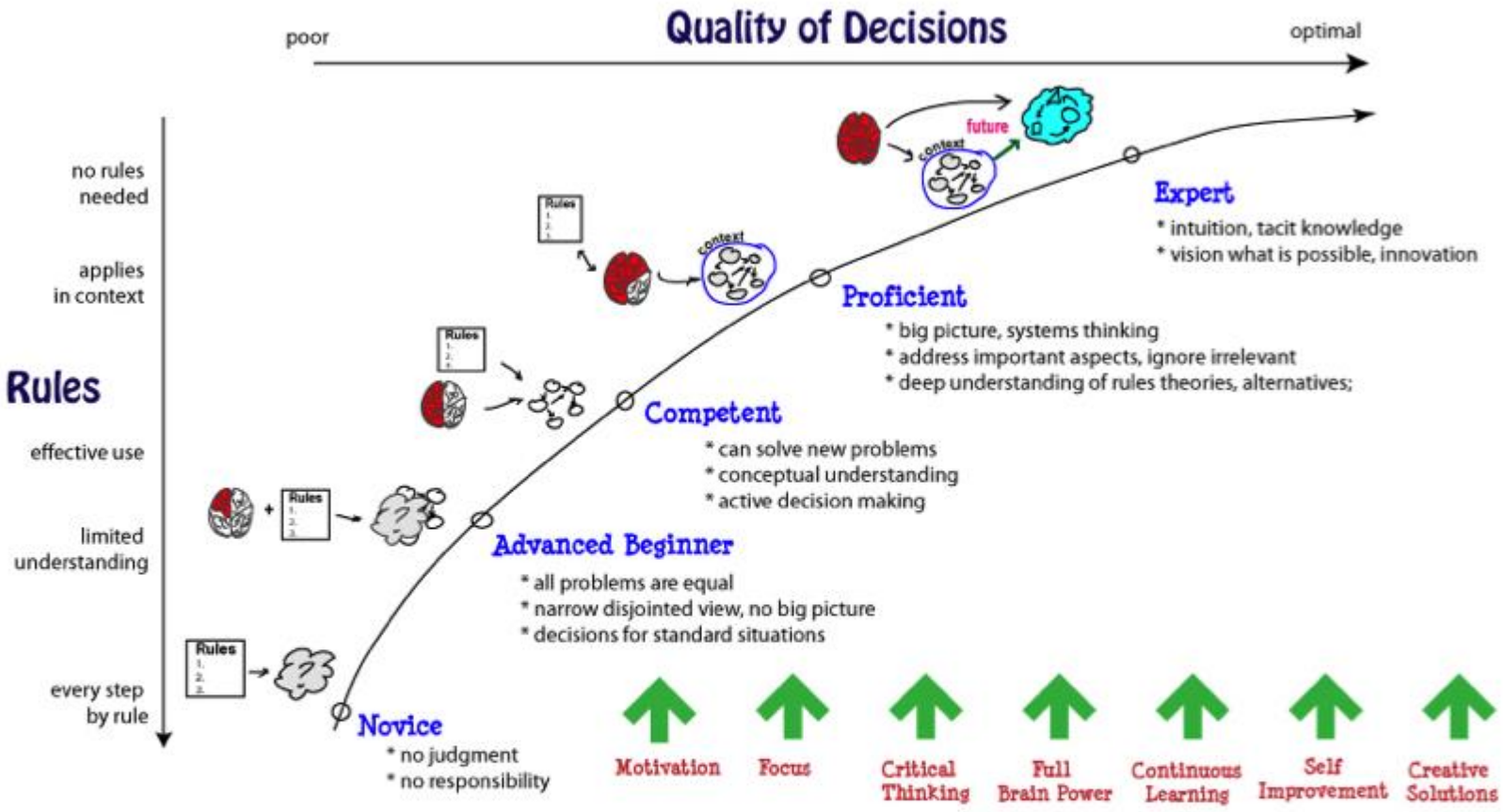
- **Novice**
  - suit aveuglément les instructions fournies
  - ne possède aucune compréhension de ce qui se passe
- **Débutant confirmé**
  - Suit des lignes directrices
  - comprend vaguement ce qui se passe autour de lui
  - Ne perçoit pas les nuances et les subtilités
- **Compétent**
  - Est autonome, prend des décisions
  - Classe, trie, ordonne ses activités et les informations
  - Perçoit les relations entre ses actions et des objectifs à long terme
  - Standardise et systématise des procédures

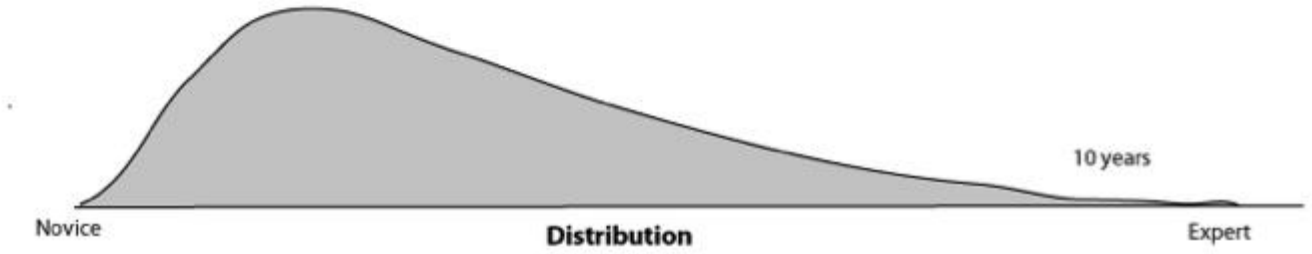
Niveaux de compétence (2/2)

- **Efficace**
  - a une vision d'ensemble et peut se détacher des détails
  - perçoit les éléments importants dans une situation et ceux qui sortent des schémas classiques
  - sa prise de décision est consciente mais son analyse est plus intuitive
  - utilise des principes pour se guider, mais peut les adapter au besoin
- **Expert**
  - a une compréhension intuitive et profonde de la situation
  - n'a plus besoin de règles, lignes directrices ou principes
  - l'approche analytique n'est utile que pour les nouvelles situations
  - fait "partie" du système
  - a une vision de ce qui est possible

Genesis of The Expert

http://gohighbrow.com/communication-thinking-and-learning/





Alors, novice, compétent ou expert ?

www.cnrs.fr

Méthodologies

Nécessité d'une approche méthodologique

www.cnrs.fr

- Construire une *bonne* application web n'est pas simple
  - interdépendance de nombreuses compétences, couches, composants
- Construire une *bonne* application web *sécurisée* est compliqué (mais pas complexe)
- Objectif compliqué ➔ décomposer en sous-objectifs plus simples
- **Méthode**: ensemble ordonné d'étapes, de principes, de règles
- **Méthodologie** : ensemble de méthodes

En sécurité, le choix d'une méthodologie est *moins* important que le fait d'en avoir au moins une

Méthodologies générales orientée sécurité

www.cnrs.fr

- Beaucoup de méthodologies existent
  - Microsoft SDL, CLASP, Touchpoint, …
- Nous allons en présenter deux
  - GISSIP (ANSSI)
  - OWASP SAMM
- Elles sont génériques et agnostiques
- Elle proposent une progression par niveau de maturité

GISSIP (ANSSI)

- « Guide d'intégration de la sécurité des systèmes d'information dans les projets »
- *Proposition* de cycle de vie de gestion de projets informatiques
  - Ce n'est ***pas*** un guide mise en œuvre PSSI !
  - Dans la logique des outils ANSSI
- Rôles et responsabilités génériques et transposables
- Organisé par phase et niveau de maturité
  - phases génériques adaptables à tous les cycles de développement
- **Homologation de sécurité**

  L'homologation de sécurité d'un SI est la déclaration par l'autorité d'homologation, conformément à une note d'orientations SSI et au vu du dossier de sécurité, que le SI considéré est apte à traiter des informations au niveau de besoins de sécurité exprimé conformément aux objectifs de sécurité, et que les risques de sécurité résiduels sont acceptés et maîtrisés.

GISSIP
Rôles

- **Maîtrise d'ouvrage**
- **Maîtrise d'œuvre**
- **Autorité d'homologation sécurité**
  - Décisionnaire en matière de sécurité
  - S'appuie sur la commission d'homologation
- **RSSI**
- **Experts techniques**
- **Comité de pilotage**
- **Commission d'homologation**
  - Un comité de pilotage orienté sécurité

GISSIP
Phases

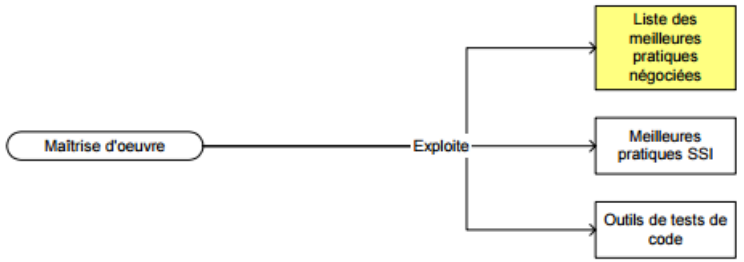
- **Etude d'opportunité**
  - prospectives, intérêt pour l'organisme
- **Etude de faisabilité**
  - faisabilité économique, organisationnelle et technique
- **Conception générale**
- **Conception détaillée**
  - réalisation
  - développement, l'intégration, la qualification et la recette
- **Exploitation**
  - déploiement, mise en œuvre, maintenance, **fin de vie**.

GISSP
Niveaux de maturité

- niveau 1
  - Usage occasionnel et informel de *best practice* SSI
- niveau 2
  - Les *best practice* sont convenablement intégrées, mais pas systématiques
- niveau 3
  - Le processus d'intégration de la SSI dans le cycle de vie des systèmes est formalisé, régulièrement utilisé
- niveau 4.
  - Le processus est devenu standard et davantage automatisé
  - *Définition et suivi d'indicateurs*,
- niveau 5.
  - Le processus d'intégration de la SSI est généralisé, bien automatisé,
  - Intégré aux processus métiers, bien accepté et s'améliore continuellement.

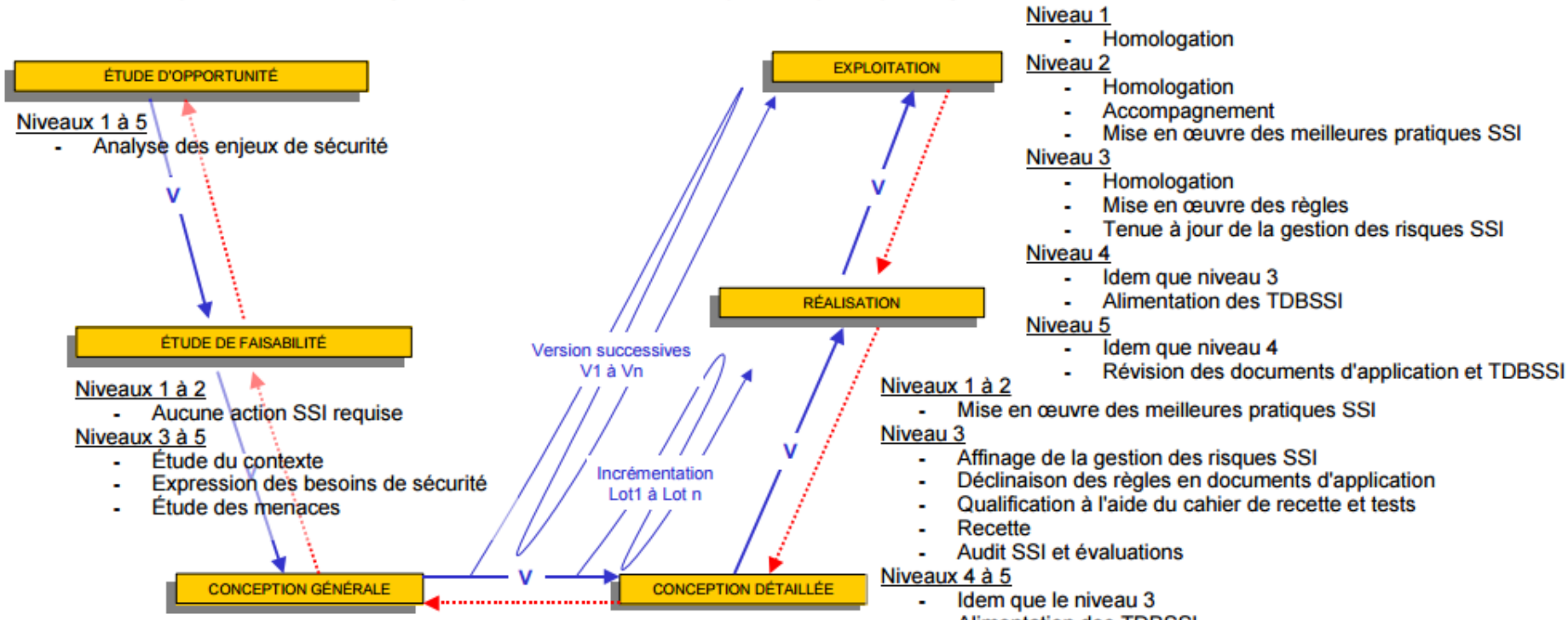
Plan d'action réalisation niveau 2 de maturité

Étape	*Étape du cycle de vie générique des SI et niveau de maturité SSI visé*
Objectif	*Objectif général de l'étape en terme de SSI*
Préalables	*Actions ou documents nécessaires en entrée de l'étape*
Description	*Description des actions SSI à mener lors de l'étape*
Livrables	*Documents livrables en sortie de l'étape*
Outils	*Outils (méthodes, catalogues…) pouvant aider à la réalisation de l'étape*
Synthèse	*Schéma présentant les principaux acteurs, outils et livrables (composant le dossier de sécurité)* 

Étape 5	Réalisation (niveau 2)
Objectif	L'objectif de la réalisation est l'application des meilleures pratiques SSI, leur vérification par la maîtrise d'œuvre et la vérification de conformité par rapport aux attentes de la maîtrise d'ouvrage.
Préalables	❑ Liste des meilleures pratiques négociées
Description	❑ Lors de la phase de **développement**, la maîtrise d'œuvre doit mettre en œuvre les meilleures pratiques SSI de conception, de développement et de gestion de configuration qui auront été retenues lors de la phase de conception détaillée. ❑ Lors de la phase de **codage**, les développeurs doivent mettre en œuvre les meilleures pratiques SSI de codage qui auront été retenues lors de la phase de conception détaillée. Ils rechercheront à mettre en œuvre les fonctions de sécurité (contrôle d'accès, filtrage…) conformément à l'état de l'art. Des tests unitaires et/ou des revues de code doivent également être réalisés. ❑ Lors de la phase d'**intégration**, les développeurs doivent mettre en œuvre les meilleures pratiques SSI d'intégration qui auront été retenues lors de la phase de conception détaillée. ❑ Lors de la phase de **qualification**, la maîtrise d'œuvre doit vérifier la bonne application des meilleures pratiques SSI retenues, vérifier le bon fonctionnement du système et confirmer les performances SSI attendues. ❑ Lors de la phase de **recette**, la maîtrise d'ouvrage doit vérifier que l'application des meilleures pratiques SSI retenues dans le cadre du développement du projet est conforme à ses attentes exprimées dans la note de stratégie de sécurité. Elle peut aussi valider la conformité de l'intégration du progiciel, ses paramétrages et ses interfaces à partir de jeux d'essai.
Livrables	❑ Aucun livrable particulier pour cette étape.
Outils	❑ Meilleures pratiques SSI ❑ Outils de tests de code
Synthèse	

4.7 Synthèse des actions SSI à mener par étape et par niveau de maturité SSI adéquat

Le schéma suivant présente de manière synthétique les actions SSI à mener pour chaque étape du cycle de vie des SI.



Niveaux 1 à 5
- Analyse des enjeux de sécurité

Niveaux 1 à 2
- Aucune action SSI requise

Niveaux 3 à 5
- Étude du contexte
- Expression des besoins de sécurité
- Étude des menaces

Niveau 1
- Aucune action SSI requise

Niveau 2
- Inventaire des meilleures pratiques SSI applicables
- Estimation de l'impact de leur application

Niveaux 3 à 5
- Identification des objectifs de sécurité

Niveau 1
- Aucune action SSI requise

Niveau 2
- Revue des choix des meilleures pratiques SSI et négociation
- Réflexion sur la nature des niveaux de garantie

Niveau 3
- Affinage de l'étude de sécurité
- Formalisation des règles de sécurité
- Précision du traitement des risques SSI

Niveaux 4 à 5
- Idem que le niveau 3
- Élaboration des TDBSSI

Niveaux 1 à 2
- Mise en œuvre des meilleures pratiques SSI

Niveau 3
- Affinage de la gestion des risques SSI
- Déclinaison des règles en documents d'application
- Qualification à l'aide du cahier de recette et tests
- Recette
- Audit SSI et évaluations

Niveaux 4 à 5
- Idem que le niveau 3
- Alimentation des TDBSSI

Niveau 1
- Homologation

Niveau 2
- Homologation
- Accompagnement
- Mise en œuvre des meilleures pratiques SSI

Niveau 3
- Homologation
- Mise en œuvre des règles
- Tenue à jour de la gestion des risques SSI

Niveau 4
- Idem que niveau 3
- Alimentation des TDBSSI

Niveau 5
- Idem que niveau 4
- Révision des documents d'application et TDBSSI

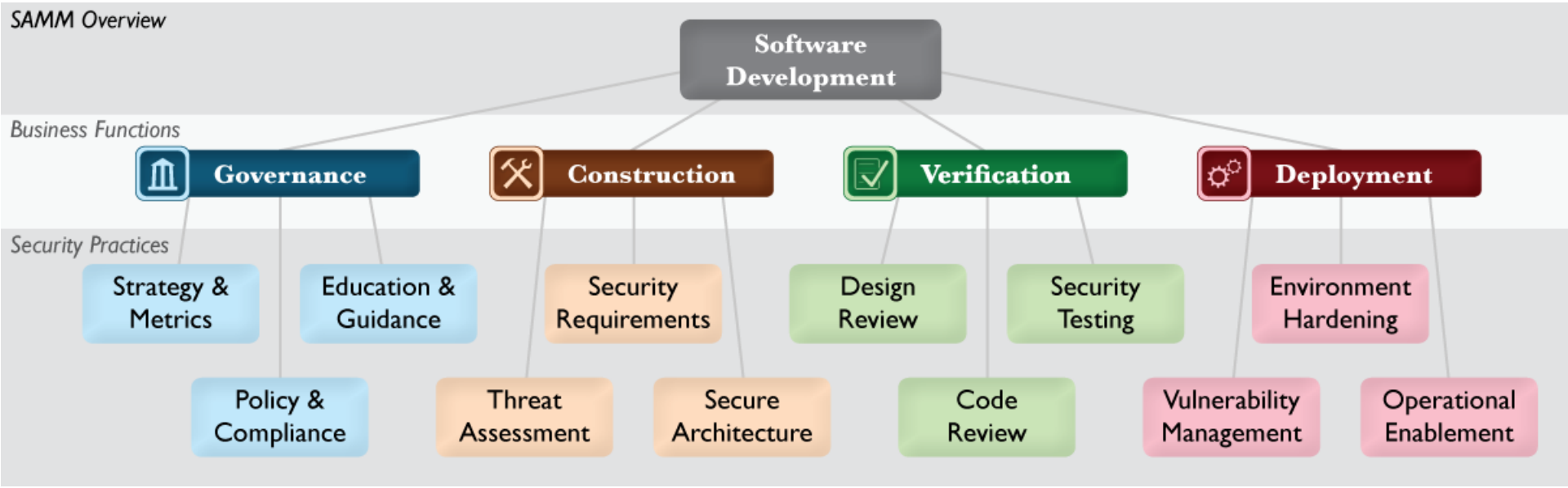
OWASP Software Assurance Maturity Model (SAMM)

www.cnrs.fr

- Guide méthodologique open source
- Modèle arborescent par niveau de maturité
- 3 niveaux de maturité seulement
  - niveau 1:
    - Compréhension initiale, et préparation à la mise en place de la *pratique*
  - niveau 2:
    - Monté en puissance et/ou *pratique* opérationnelle
  - niveau 3:
    - Maitrise complète de la *pratique*

OWASP SAMM

- Arborescence
  - *Fonction*
    - *Pratiques de sécurité*
      - *Objectifs*



Objectif SAMM (1/2)

- Chaque objectif
  - Des tâches
  - Des prérequis
  - Des résultats
  - Des mesures
  - Des coûts
  - Des rôles
  - Des niveaux relatifs

Secure Architecture

	SA 1	SA 2	SA 3
Objective	Insert consideration of proactive security guidance into the software design process	Direct the software design process toward known-secure services and secure-by-default designs	Formally control the software design process and validate utilization of secure components
Activities	A. Maintain list of recommended software frameworks B. Explicitly apply security principles to design	A. Identify and promote security services and infrastructure B. Identify security design patterns from architecture	A. Establish formal reference architectures and platforms B. Validate usage of frameworks, patterns, and platforms
Assessment	✦ Are project teams provided with a list of recommended third-party components? ✦ Are project teams aware of secure design principles and do they apply them consistently?	✦ Do you advertise shared security services with guidance for project teams? ✦ Are project teams provided with prescriptive design patterns based on their application architecture?	✦ Do project teams build software from centrally-controlled platforms and frameworks? ✦ Are project teams audited for the use of secure architecture components?
Results	✦ Ad hoc prevention of unexpected dependencies and one-off implementation choices ✦ Stakeholders aware of increased project risk due to libraries and frameworks chosen ✦ Established protocol within development for proactively applying security mechanisms to a design	✦ Detailed mapping of assets to user roles to encourage better compartmentalization in design ✦ Reusable design building blocks for provision of security protections and functionality ✦ Increased confidence for software projects from use of established design techniques for security	✦ Customized application development platforms that provide built-in security protections ✦ Organization-wide expectations for proactive security effort in development ✦ Stakeholders better able to make tradeoff decisions based on business need for secure design

Objectif SAMM (2/2)

Secure Architecture

SA 1

Cible

Insert consideration of proactive security guidance into the software design process

Description détaillée des activités

Activities

A. Maintain list of recommended software frameworks

Across software projects within the organization identify commonly used third-party software libraries and frameworks in use. Generally, this need not be an exhaustive search for dependencies, but rather focus on capturing the high-level components that are most often used.

From the list of components, group them into functional categories based on the core features provided by the third-party component. Also, note the usage prevalence of each component across project teams to weight the reliance upon the third-party code. Using this weighted list as a guide, create a list of components to be advertised across the development organization as recommended components.

Several factors should contribute to decisions for inclusion on the recommended list. Although a list can be created without conducting research specifically, it is advisable to inspect each for incident history, track record for responding to vulnerabilities, appropriateness of functionality for the organization, excessive complexity in usage of the third-party component, etc.

This list should be created by senior developers and architects, but also include input from managers and security auditors. After creation, this list of recommended components matched against functional categories should be advertised to the development organization. Ultimately, the goal is to provide well-known defaults for project teams.

B. Explicitly apply security principles to design

During design, technical staff on the project team should use a short list of guiding security principles as a checklist against detailed system designs. Typically, security principles include defense in depth, securing the weakest link, use of secure defaults, simplicity in design of security functionality, secure failure, balance of security and usability, running with least privilege, avoidance of security by obscurity, etc.

In particular for perimeter interfaces, the design team should consider each principle in the context of the overall system and identify features that can be added to bolster security at each such interface. Generally, these should be limited such that they only take a small amount of extra effort beyond the normal implementation cost of functional requirements and anything larger should be noted and scheduled for future releases.

While this process should be conducted by each project team after being trained with security awareness, it is helpful to incorporate more security-savvy staff to aide in making design decisions.

Prérequis

Résultats

Métriques

Coûts

Compétences

Assessment

- Are project teams provided with a list of recommended third-party components?
- Are project teams aware of secure design principles and do they apply them consistently?

Results

- Ad hoc prevention of unexpected dependencies and one-off implementation choices
- Stakeholders aware of increased project risk due to libraries and frameworks chosen
- Established protocol within development for proactively applying security mechanisms to a design

Success Metrics

- >80% of development staff briefed on software framework recommendations in past 1 year
- >50% of projects self-reporting application of security principles to design

Costs

- Buildout, maintenance, and awareness of software framework recommendations
- Ongoing project overhead from analysis and application of security principles

Personnel

- Architects
- Developers
- Security Auditors
- Managers

Related Levels

- Education & Guidance - 1

TLDR;

- GISSIP
  - Accessible, très bien rédigé,
  - *En français*.
  - Cohérence ANSII
  - Il est complété par d'autres guides de bonnes pratiques
  - Un peu vieux (2006)
  - Un peu vieux jeu ☺
  - Pas de solution clef-en-main, il reste assez abstrait.
  - Documentation sous forme PDF seulement
- OWASP SAMM
  - Très complet, ouvert et international,
  - *En anglais*
  - Actuel
  - Vivant
    - version 2.0 en préparation
  - Beaucoup d'outils associés
  - Profiling de charge par activité (*cf.* How-To)
  - Documentation sous plusieurs formats
    - Online, PDF, livre

Evaluation du besoin de maturité SSI

www.cnrs.fr

- Questionnaires d'évaluation de la maturité nécessaire à votre organisation
  - OpenSAMM_Assessment_Toolbox
  - Guide relatif à la maturité SSI (ANSSI)
- Survol du questionnaire ANSII en 12 points
  - Les points abordés doivent être évalués sur une échelle de 0 à 4
  - Objectif, mesuré … et instructif

Questionnaire (1/2)

- Les conséquences potentielles
  - **Adhérence au système d'information (SI)** : importance pour vos missions ?
    - *accessoire, utile, nécessaire, vitale*
  - **Niveau des impacts internes** : conséquences internes d'un sinistre SSI ?
    - *négligeables, significatives, graves, fatales*
  - **Niveau des impacts externes** : (…externes…)
    - *négligeables, significatives, graves, fatales*

- La sensibilité du patrimoine
  - **Défaillance disponibilité** : *négligeable, perturbante, grave, fatale*
  - **Défaillance d'intégrité** : …
  - **Défaillance de confidentialité** : …

Questionnaire (2/2)

- Le degré d'exposition aux menaces
  - **Fréquences des sinistres**:
    - *rarissimes, plusieurs par année, par trimestre, par mois*
  - **Degré de motivation des attaquants** (potentiels) :
    - *attaque improbable, motivation faible, forte, très importante*
  - **Moyens des attaquants** (potentiels):
    - *faible, significatifs, importants, potentiellement illimités*

- L'importance des vulnérabilités
  - **Hétérogénéité du SI** :
    - *homogène, hétérogénéité faible, forte, extrêmement forte*
  - **Ouverture du SI:**
    - *Non ouvert, ouvert systèmes interne, externe sous contrôle, hors contrôle*
  - **Variabilité du SI**
    - *Contexte stable, changements rares, assez souvent, très souvent*

N’ayez pas peur…

www.cnrs.fr

- Même si ces méthodologies semblent effrayantes et peu engageantes
- Avoir une vision de la SSI
  - au delà du simple développement
  - à un niveau organisationnel
- Entamer une réflexion de tous les acteurs,
  - de la direction du service à l’exploitation.

Application au cycle de développement logicielle

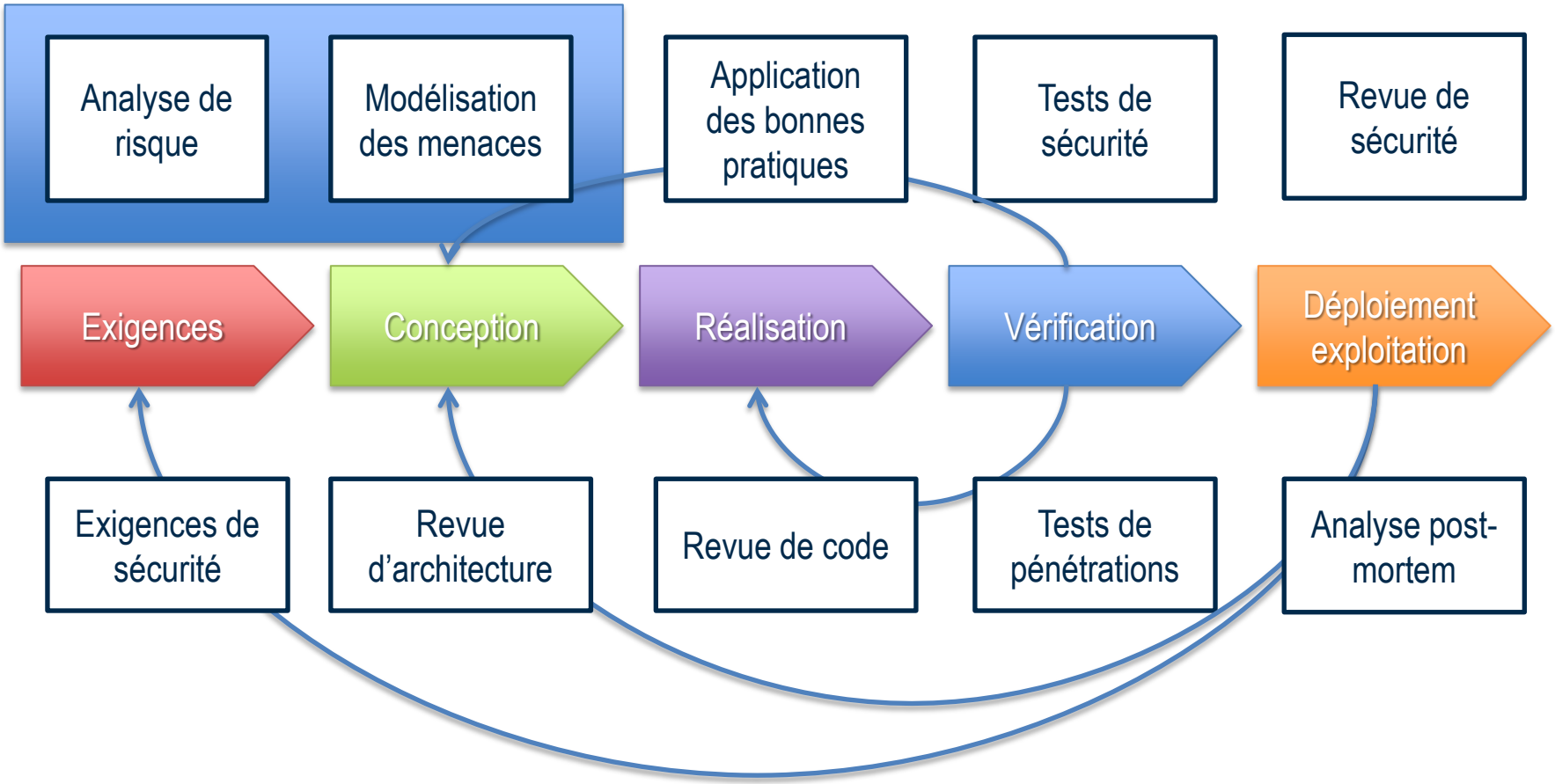
Lequel est le meilleur ?

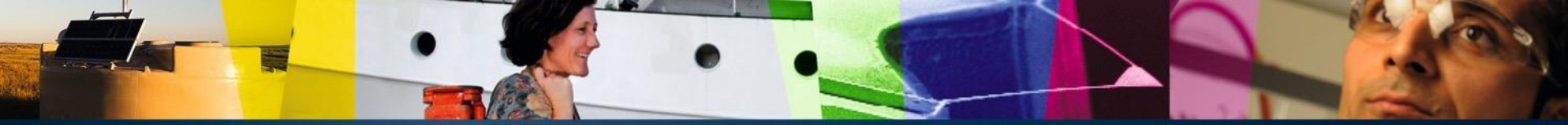
Cycle de développement

- Le bon outil pour le bon usage
  - https://fr.wikipedia.org/wiki/Cycle_de_d%C3%A9veloppement_(logiciel)
- Choisissez le votre :
  - Des situations **claires**
    - où les contrôles de sécurité peuvent s'insérer (analyse de risque, revue de code, …)
  - Une bonne adéquation avec la taille et la maturité de l'organisation
  - qui (quand même) peut réduire le taux d'erreur et augmenter la productivité
  - que vous **voulez**/**pouvez** appliquer vraiment

- **En 2017 pouvons-nous encore avoir le luxe de ne pas en avoir ?**

Exemple simpliste





Références

www.cnrs.fr

- Modèle de Dreyfuss et Dreyfuss
  - https://media.pragprog.com/titles/ahptl/chap2.pdf
  - http://www.doceo.co.uk/background/expertise.htm
  - http://www.skorks.com/2009/08/building-software-development-expertise-using-the-dreyfus-model/
  - http://gohighbrow.com/communication-thinking-and-learning/