Modèles de maturité

Louis Nadeau

Bentley AppSec Team

© 2017 Bentley Systems, Incorporated

Qui suis-je?

- Louis Nadeau
- Directeur de la sécurité applicative chez Bentley Systems
- (Aussi Vice-Leader OWASP Québec)

- Louis.Nadeau@Bentley.com

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Sujets

- Intro
- Modèles de maturité
  - OWASP SAMM 1.5
  - Comparaison avec d’autres cadres & standards
- Autres considérations
  - Agile et « Security stories »
  - Réseau de champions
  - DevOps et DevSecOps
- Conclusion

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

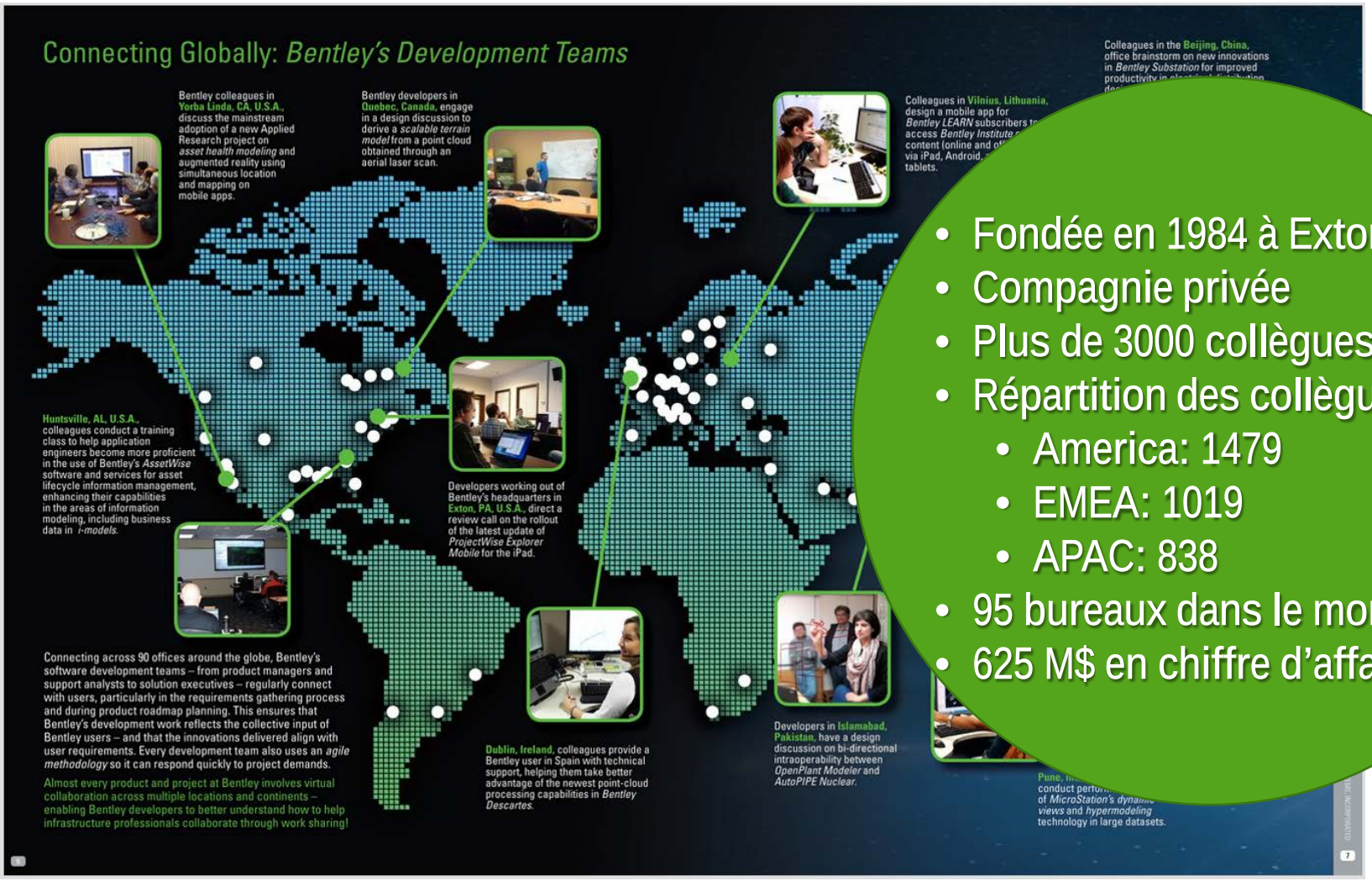
Bentley Systems

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

La mission de Bentley est de développer des ***logiciels et des services innovants*** pour les entreprises et les professionnels qui ***conçoivent, construisent et opèrent*** le monde de l’infrastructure afin de soutenir l'économie mondiale et l’environnement pour une meilleure qualité de vie.

Un leader mondial



- Fondée en 1984 à Exton (USA)
- Compagnie privée
- Plus de 3000 collègues
- Répartition des collègues:
  - America: 1479
  - EMEA: 1019
  - APAC: 838
- 95 bureaux dans le monde
- 625 M$ en chiffre d'affaires

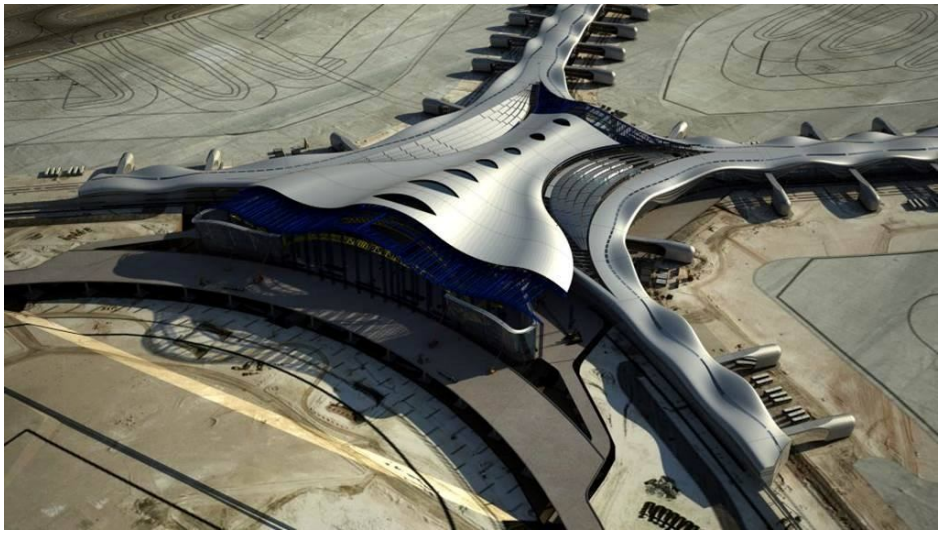
≈300 produits

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Bentley System

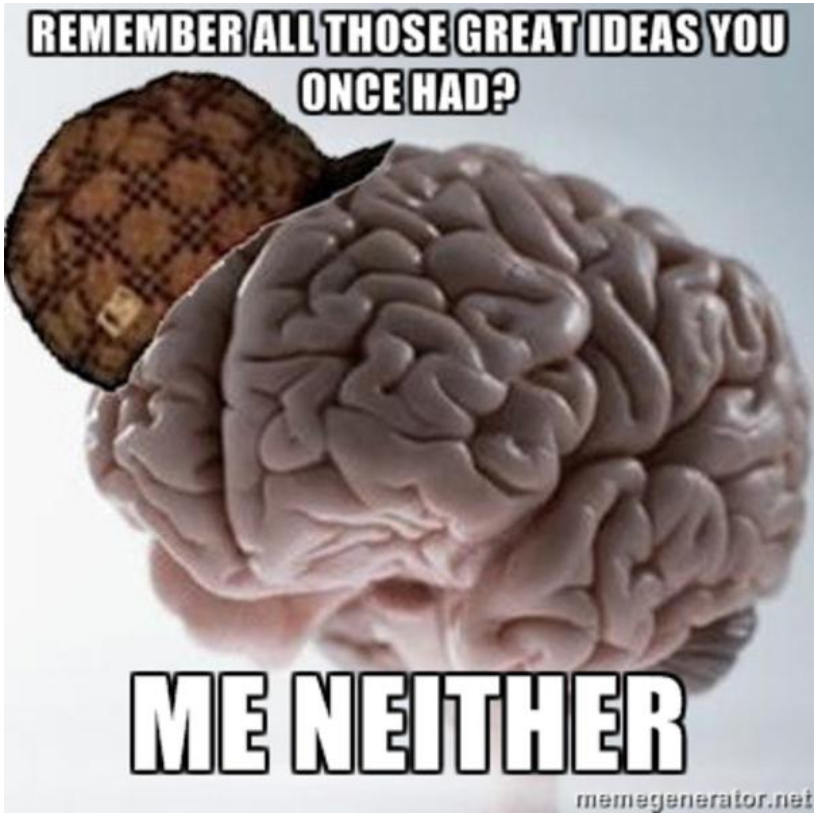
Bentley System



 © 2017 Bentley Systems, Incorporated

Introduction & Motivation

- Vous avez appris plein de choses aujourd'hui
  - Par où commencer?
  - Comment justifier votre choix?



 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Modèles de maturité: OWASP SAMM

 © 2017 Bentley Systems, Incorporated

Bentley® AppSec Team

Qu’est-ce qu’un modèle de maturité?

- « Framework that help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization. »
  - Tiré du guide OWASP SAMM

- Aide à prendre des décisions rationnelles qui ne sont pas basées sur: le HDD (« Hype Driven Development») et ses variantes
  - Reddit Driven Development
  - Conference Driven Development
  - Loudest Guy Driven Decision

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

OWASP SAMM 1.5

- « Software Assurance Maturity Model (SAMM) »

- Historique
  - 2008 : SAMM Beta (Consultant + Fortify)
  - 2009 : OpenSAMM 1.0
  - (Fork)
  - 2017 : OWASP SAMM 1.5
  - 2018 : OWASP SAMM 2.0 (en cours de développement)

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Applicabilité des modèles de maturité

- Quand un modèle de maturité est-il utile?
  - S'appuyer sur un cadre crédible pour orienter ou justifier des améliorations
- Modèle de maturité ≠ certification
  - Un modèle n'est pas une certification, souvent ils se supportent mutuellement
- Modèle de maturité ≠ cadre légal
- Modèle de maturité ≠ normes de programmation

 © 2017 Bentley Systems, Incorporated

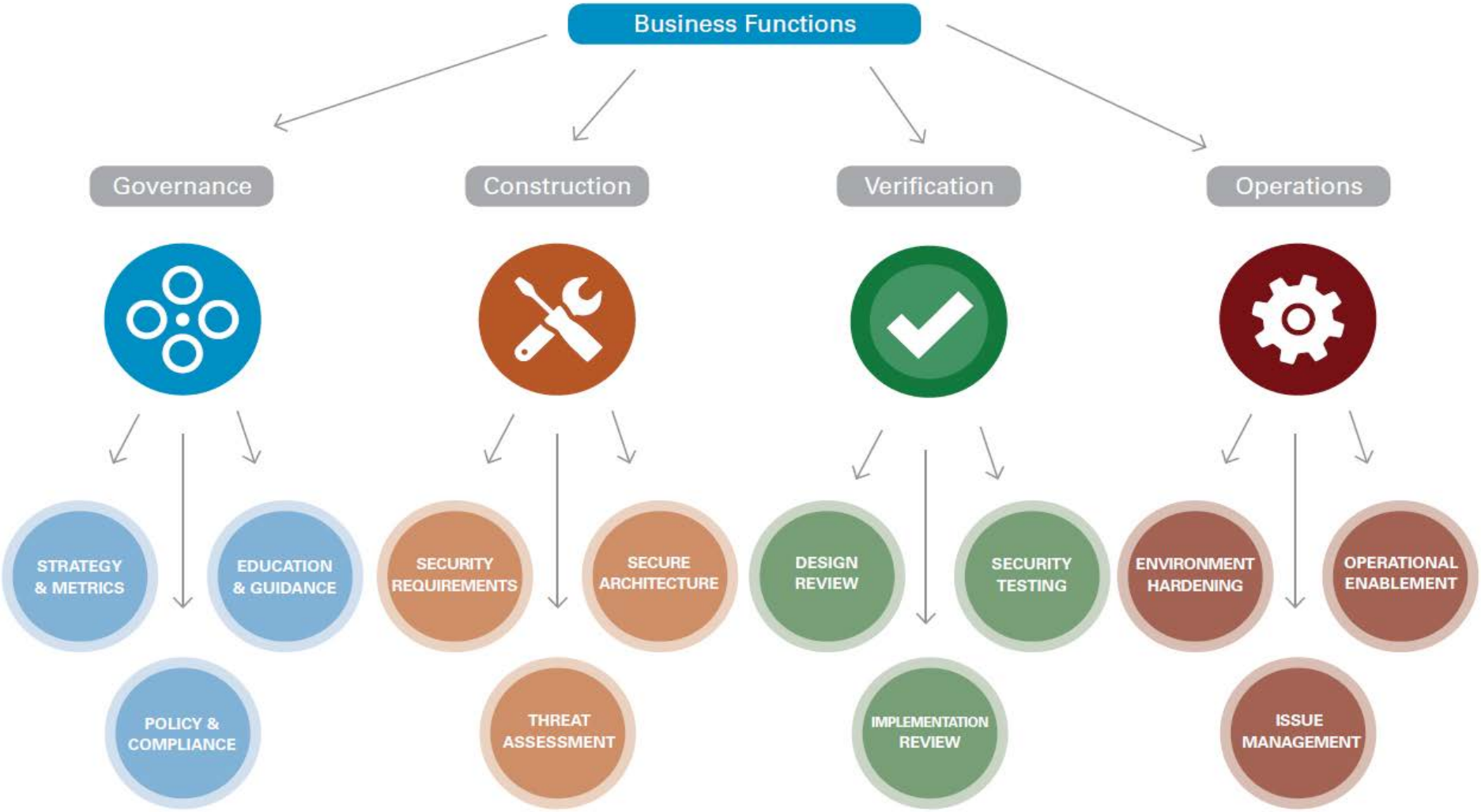
Bentley AppSec Team

Applicabilité des modèles de maturité

- Est-ce que c'est coûteux?
  - Non, c'est gratuit et rapide
- Un questionnaire et des documents gratuits
- Vous prenez ce que vous voulez
  - Pas de police (i.e. auditeur) du modèle de maturité
- Peut être appliqué sur une organisation, une partie de l'organisation, un projet, etc.

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team



 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

OWASP SAMM 1.5 : Comment ça marche?

1. Faire une évaluation de la maturité courante
   - Questionnaire
   - Donne une note sur 3 dans les différentes catégories
2. Définir la stratégie (objectifs)
   - Vous n'en avez pas vraiment besoin, on peut utiliser les chartes fournies par OWASP. C'est ici que la crédibilité est « injectée ».
3. Formuler un plan
   - En utilisant les plans prédéfinis, la seule chose à faire c'est de décider la longueur des phases : semaines, mois ou années.
4. Implémenter le plan
   - Plein de recommandations intéressantes dans le document fourni

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

(1) Évaluation

	B	C	D	E	I	J
136	Construction					
137	Threat Assessment			Answer	Notes	Rating
138	TA1	Do projects in your organization consider and document likely threats?		Yes, a small percentage		1.10
139		*Guidance:* Likely worst-case scenarios are documented for each project based on its business risk profile.				
140		*Guidance:* Attack trees or a threat model is created for each project tracing the preconditions necessary for a worst-case				
141		*Guidance:* Attack trees or threat models are expanded to include potential security failures in current and historical				
142		*Guidance:* When new features are added to a project, attack trees or threat models are updated.				
143						
144		Does your organization understand and document the types of attackers it faces?		Yes, a small percentage		
145		*Guidance:* Potential external threat agents and their motivations are documented for each project.				
146		*Guidance:* Potential internal threat agents, their associated roles, and damage potential are documented for each project or				
147		*Guidance:* A common set of threat agents, motivations, and other information is collected at the organization level and re-				
148						
149						
150	TA2	Do project teams regularly analyze functional requirements for likely abuses?		Yes, a small percentage		
151		*Guidance:* Each project derives abuse-cases from its use-cases.		No		
152		*Guidance:* As project requirements or features are added, abuse-cases are updated.		Yes, a small percentage are/do		
153				Yes, at least half of them are/do		
154		Do project teams use a method of rating threats for relative comparison?		Yes, the majority of them are/do Yes, a small percentage		
155		*Guidance:* A documented weighting system based on documented threat agents, exploit value, technical difficulty, and other factors is used to rank threats.				
156		*Guidance:* Remediation of vulnerabilities is prioritized based on the weighting system.				
157						
158		Are stakeholders aware of relevant threats and ratings?		Yes, at least half of them		
159		*Guidance:* Potential threats and ratings are reviewed with project stakeholders.				
160						
161						
162	TA3	Do project teams specifically consider risk from external software?		Yes, the majority of them		
163		*Guidance:* Third-party, external libraries and code used in each project are clearly identified and documented for each				
164		*Guidance:* Project threat models are updated based on identified threat agents and motivations for third party libraries and				
165						
166		Are the majority of the protection mechanisms and controls captured and mapped back to threats?		Yes, a small percentage		
167		*Guidance:* An assessment for each project has been conducted to identify mitigating controls that prevent preconditions identified in attack trees or threat models.				
168		*Guidance:* This assessment is updated each time new features or requirements are introduced or the attack tree is				
169		*Guidance:* Mitigating controls have been documented within the attack tree or threat model				

 © 2017 Bentley Systems, Incorporated

(1) Évaluation

Current Maturity Score					
			Maturity		
Functions	Security Practices	Current	1	2	3
Governance	Strategy & Metrics	3.00	1.00	1.00	1.00
Governance	Policy & Compliance	1.85	0.75	0.50	0.60
Governance	Education & Guidance	1.85	0.10	1.00	0.75
Construction	Threat Assessment	1.45	0.35	0.50	0.60
Construction	Security Requirements	0.50	0.20	0.20	0.10
Construction	Secure Architecture	0.90	0.35	0.35	0.20
Verification	Design Analysis	1.25	0.50	0.25	0.50
Verification	Implementation Review	1.35	0.35	0.50	0.50
Verification	Security Testing	1.67	0.67	0.50	0.50
Operations	Issue Management	1.58	0.23	0.60	0.75
Operations	Environment Hardening	0.30	0.10	0.10	0.10
Operations	Operational Enablement	1.35	0.35	0.50	0.50

Functions	Current
Governance	2.23
Construction	0.95
Verification	1.42
Operations	1.08

 © 2017 Bentley Systems, Incorporated

(2) Définir les objectifs

- Définir le pointage final qu'on veut obtenir pour les 12 pratiques
- La plupart des gens vont utiliser un patron déjà existant
  - Ça donne de la crédibilité de s'appuyer sur ce qu'OWASP recommande pour votre industrie/produit

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

(2) Définir les objectifs

INDEPENDENT SOFTWARE VENDOR

ROADMAP TEMPLATE

Rationale

An independent software vendor involves the core business function of building and selling software components and applications.

Initial drivers to limit common vulnerabilities affecting customers and users leads to early concentration on implementation review and security testing activities.

Shifting toward more proactive prevention of security errors in product specification, an organization adds activities for security requirements over time.

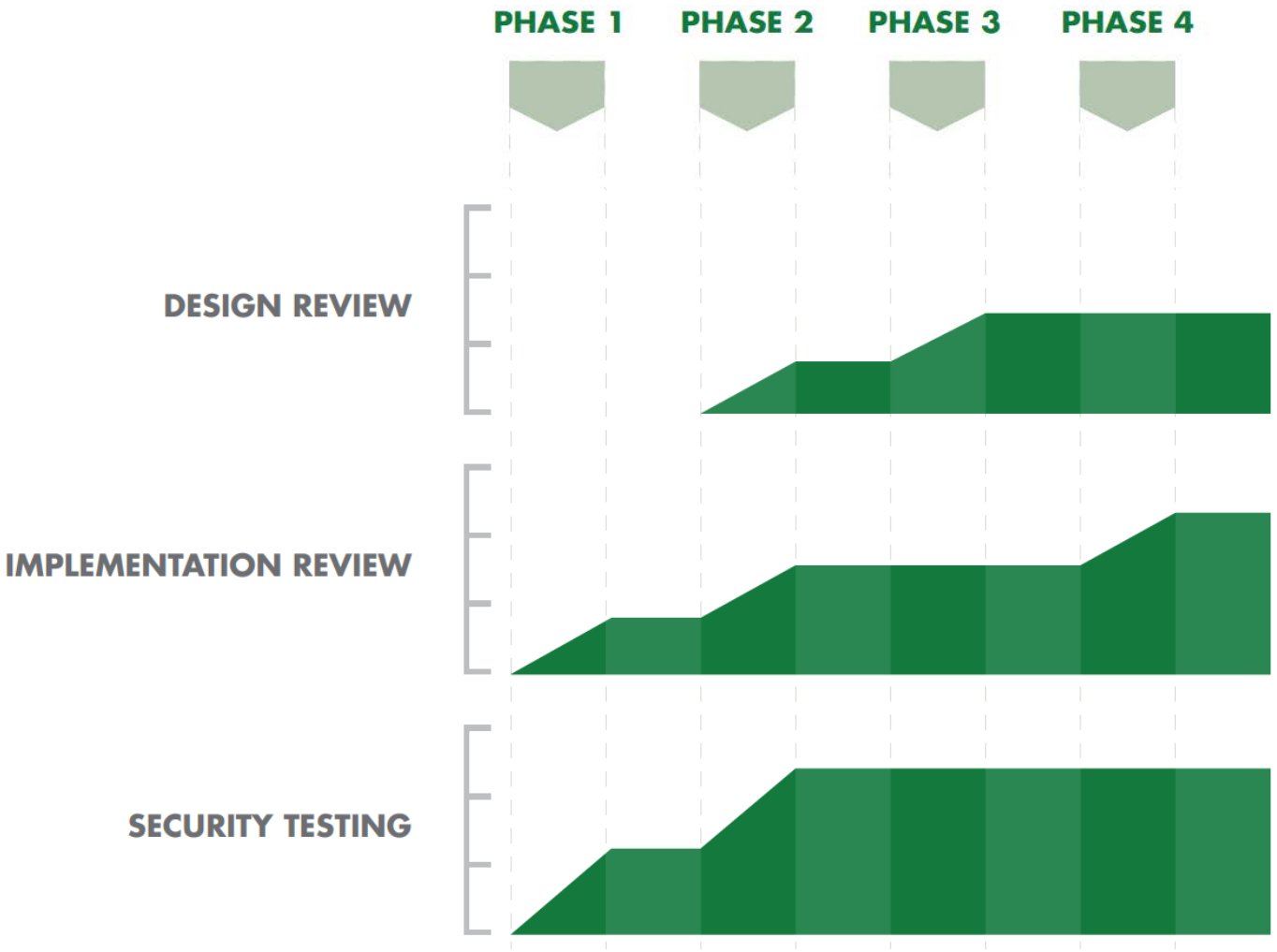
Also, to minimize the impact from any discovered security issues, the organization ramps up issue management activities over time.

As the organization matures, knowledge transfer activities from operational enablement are added to better inform customers and users about secure operation of the software.

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

(2) Définir les objectifs

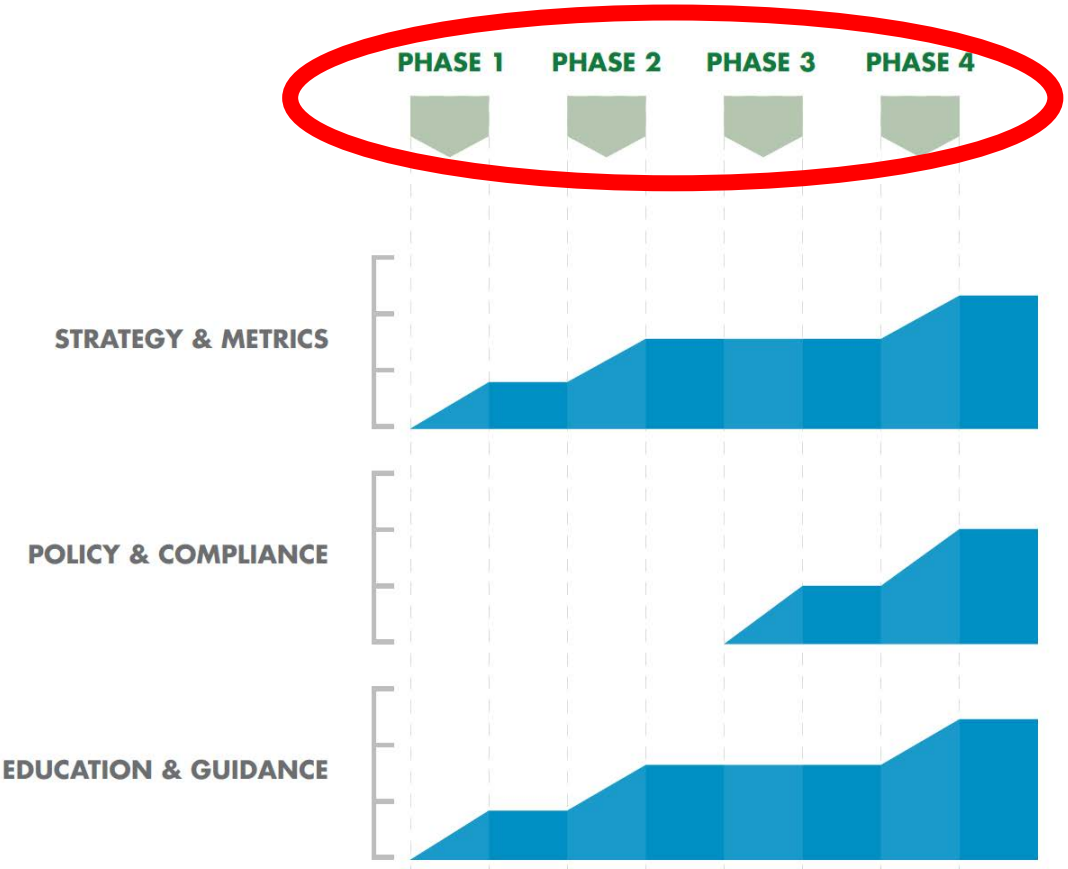


 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

(3) Formuler un plan

- De quelle longueur seront vos phases?
  - Plus important : quelle est la longueur que vous pouvez vous permettre?



 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

(4) Implémenter le plan

- Activités (niveau 1):
  - Créer des tests à partir des requis de sécurité
  - Faire des tests de pénétration au moment de la sortie du logiciel
- Évaluations, résultats, métriques, coûts, personnel et niveaux reliés

Security Testing

ST 1

Establish process to perform basic security tests based on implementation and software requirements

Activities

A. Derive test cases from known security requirements

From the known security requirements for a project, identify a set of test cases to check the software for correct functionality. Typically, these test cases are derived from security concerns surrounding the functional requirements and business logic of the system, but should also include generic tests for common vulnerabilities based on the implementation language or technology stack.

Often, it is most effective to use the project team's time to build application-specific test cases and utilize publicly available resources or purchased knowledge bases to select applicable general test cases for security. Although not required, automated security testing tools can also be utilized to cover the general security test cases.

This test case planning should occur during the requirements and/or design phases, but must occur before final testing prior to release. Candidate test cases should be reviewed for applicability, efficacy, and feasibility by relevant development, security, and quality assurance staff.

B. Conduct penetration testing on software releases

Using the set of security test cases identified for each project, penetration testing should be conducted to evaluate the system's performance against each case. It is common for this to occur during the testing phase prior to release.

Penetration testing cases should include both application-specific tests to check soundness of business logic as well as common vulnerability tests to check the design and implementation. Once specified, security test cases can be executed by security-savvy quality assurance or development staff, but first-time execution of security test cases for a project team should be monitored by a security auditor to assist and coach team members.

Prior to release or deployment, stakeholders must review results of security tests and accept the risks indicated by failing security tests at release time. In the latter case, a concrete timeline should be established to address the gaps over time.

Assessment

- Do projects specify security testing based on defined security requirements
- Is penetration testing performed on high risk projects prior to release?
- Are stakeholders aware of the security test status prior to release?

Results

- Independent verification of expected security mechanisms surrounding critical business functions
- High-level due diligence toward security testing
- Ad hoc growth of a security test suite for each software project

Success Metrics

- >50% of projects specifying security test cases in the past 12 months
- >50% of stakeholders briefed on project status against security tests in the past six months

Costs

- Buildout or license of security test case
- Ongoing project overhead from maintenance and evaluation of security test cases

Personnel

- QA Testers
- Security Auditor
- Developers
- Architects
- Business Owners

Related Levels

- Security Requirements - 1

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

(4) Implémenter le plan

- Activités (niveau 3):
  - Utiliser des outils de test spécifiques à l'application pour faire des tests de sécurité
  - Définir des bornes tout au long du processus de développement pour faire des tests de sécurité

Security Testing

ST 3

Require application-specific security testing to ensure baseline security before deployment

Activities

A. Employ application-specific security testing automation

Through either customization of security testing tools, enhancements to generic test case execution tools, or buildout of custom test harnesses, project teams should formally iterate through security requirements and build a set of automated checkers to test the security of the implemented business logic.

Additionally, many automated security testing tools can be greatly improved in accuracy and depth of coverage if they are customized to understand more detail about the specific software interfaces in the project under test. Further, organization-specific concerns from compliance or technical standards can be codified as a reusable, central test battery to make audit data collection and per-project management visibility simpler.

Project teams should focus on buildout of granular security test cases based on the business functionality of their software, and an organization-level team led by a security auditor should focus on specification of automated tests for compliance and internal standards.

B. Establish release gates for security testing

To prevent software from being released with easily found security bugs, a particular point in the software development lifecycle should be identified as a checkpoint where an established set of security test cases must pass in order to make a release from the project. This establishes a baseline for the kinds of security tests all projects are expected to pass.

Since adding too many test cases initially can result in an overhead cost bubble, begin by choosing one or two security issues and include a wide variety of test cases for each with the expectation that no project may pass if any test fails. Over time, this baseline should be improved by selecting additional security issues and adding a variety of corresponding test cases.

Generally, this security testing checkpoint should occur toward the end of the implementation or testing, but must occur before release.

For legacy systems or inactive projects, an exception process should be created to allow those projects to continue operations, but with an explicitly assigned timeframe for mitigation of findings. Exceptions should be limited to no more that 20% of all projects.

Assessment

- Are security test cases comprehensively generated for application-specific logic?
- Does a minimum security baseline exist for security testing?

Results

- Organization-wide baseline for expected application performance against attacks
- Customized security test suites to improve accuracy of automated analysis
- Project teams aware of objective goals for attack resistance

Success Metrics

- >50% of projects using security testing customizations
- >75% of projects passing all security tests in the past six months

Costs

- Buildout and maintenance of customizations to security testing automation
- Ongoing project overhead from security testing audit process
- Organization overhead from project delays caused by failed security testing audits

Personnel

- Architects
- Developers
- Security Auditors
- QA Testers
- Business Owners
- Managers

Related Levels

- Policy & Compliance - 2
- Secure Architecture - 3

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

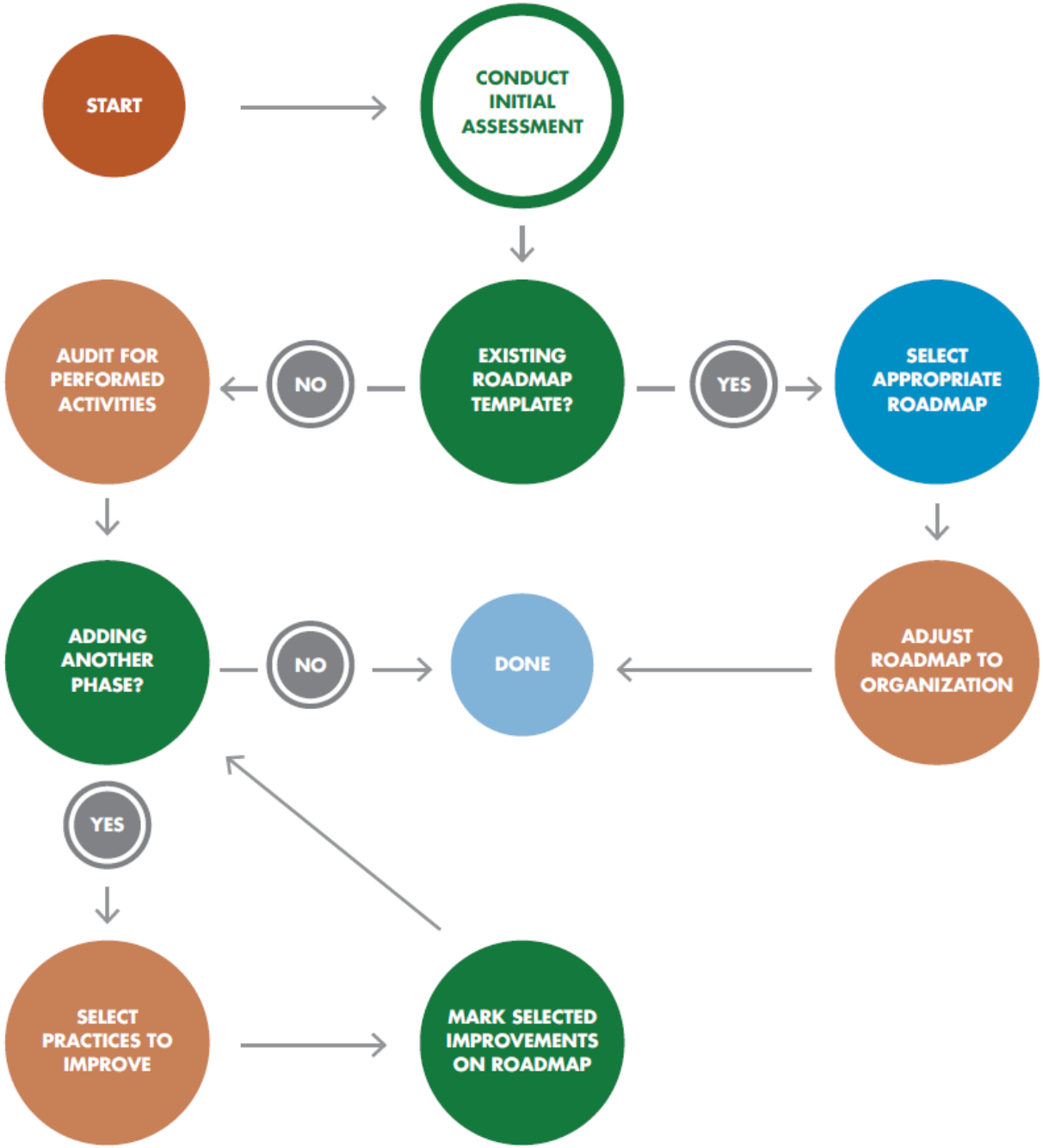
TL;DR OWASP SAMM

On fait quoi demain?

1. Répondre aux questions et trouver le pointage courant
2. Choisir un patron qui vous convient
3. Décider de la durée des phases
4. Commencer à implémenter ce qu'il manque pour se rendre au prochain niveau

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team



Bentley AppSec Team

Comparaison avec d’autres cadres & standards

- Modèles :
  - **Microsoft SDL**
  - BSIMM
  - CLASP
  - Touchpoints
  - SSE-CMM
  - C2M2
- Certifications :
  - **ISO27034**
  - ISO27001
  - SOC2
  - FedRAMP

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

OWASP SAMM Vs Microsoft SDL

	OWASP SAMM	Microsoft
Objectifs	Dépend du logiciel développé	Atteindre le niveau « Dynamic »
Trajectoires	4 niveaux indépendants, la trajectoire est différente pour chaque projet	4 niveaux et une trajectoire fixe
Outils	Test d'autoévaluation, guides et patrons	4 livres de base. Un livre pour Agile et un papier récent pour DevOps. Template Visual Studio
Biais	Web	Desktop

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

OWASP SAMM Vs Microsoft SDL

Training, Policy and Organization	Basic	Training needs have been determined and content and an appropriate curriculum for developers and testers in the organization has been created or acquired
Training, Policy and Organization	Basic	Architects, developers, and testers for Basic to Standardized pilot projects have completed introductory training.

Governance	Education & Guidance	1	Have most developers been given high-level security awareness training ?
Governance	Education & Guidance	1	Does each project team have access to secure development best practices and guidance?
Governance	Education & Guidance	2	Are most roles in the development process given role-specific training and guidance?

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Autres considérations
(mon opinion sur ce qu'il manque aux modèles de maturité aujourd'hui)

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

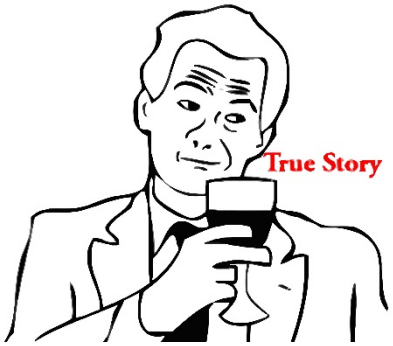
Notes

- OWASP SAMM 2 est supposé sortir fin 2018
  - Le brouillon vient juste de sortir:
  - https://github.com/OWASP/samm/tree/master/v2.0

 © 2017 Bentley Systems, Incorporated

Agile

- Vous avez appris des choses aujourd'hui
  - Vous voulez commencer demain…
  - Mais on est dans le milieu du sprint…
  - Quoi faire?
- La sécurité vient toujours déranger vos développeurs avec des bogues urgents pendant le sprint et ça brise votre planif…
  - Quoi faire?
- « Security Stories »!!



 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Agile « Security Stories »

- Qu’est-ce qu’une « security story »?

- Avantages :
  - Prioriser la sécurité comme le reste du « backlog »
  - Standardiser l’implémentation
  - Réduire le « jeu du téléphone »
  - Sauver du temps : faire la recherche une seule fois
  - Sauver de l’argent : faire la sécurité plus tôt dans le cycle du développement
    - « Story » tôt au lieu de bogue plus tard

- SAFEcode Security Stories
  - http://www.safecode.org/publication/SAFECode_Agile_Dev_Security0712.pdf

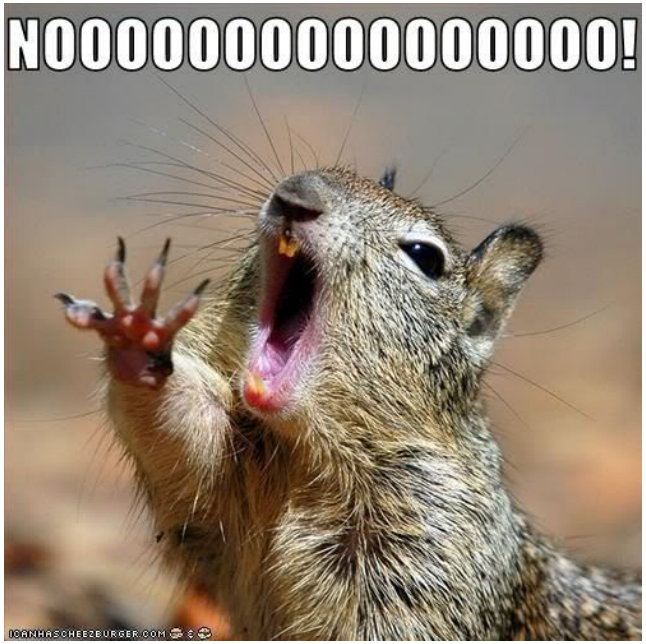
- Agile est mentionnée une seule fois dans les documents d’OWASP SAMM…

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Agile

- La plupart des organisations commencent leur sécurité applicative avec la méthode immature de faire un « pentest » de temps en temps quand on est prêt à faire un déploiement
  - Il ne faut pas déranger les développeurs pendant un sprint
    - **Surtout pas le dernier!!**



 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

SAFEcode Security Stories

No.	Security-focused story	Backlog task(s)	SAFECode Fundamental Practice(s)	CWE-ID
18	As a(n) architect/developer, I want to ensure **AND** as QA, I want to verify that cross-site request forgery attacks are prevented	**[D]** Use one of the many available libraries and frameworks that takes CSRF into account. **[D]** Defend against cross-site scripting (see Story 17). **[A/D]** Add business logic and workflow steps to critical processes in the system, and make them out-of-band: send an email in case of password change, send a text message when changing a critical value. **[D/T]** Log critical operations and the details of their initiation and arguments. **[A/D]** Do not use HTTP GET for any method that effects a change in system state.	• Use Anti-Cross Site Scripting (XSS) Libraries • Validate Input and Output to Mitigate Common Vulnerabilities • Use Logging and Tracing	CWE-352

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Exemple personnalisé

- **[Story] Implement CORS headers**
  - **[Task] Set the CORS headers**
    - Access-Control-Expose-Headers
    - Access-Control-Allow-Credentials
    - Access-Control-Allow-Origin
      - Note: Do not set Access-Control-Allow-Origin to "null"
      - Use the Access-Control-Allow-Origin header only on chosen URLs that need to be accessed cross-domain.
      - Note: Do not use the Access-Control-Allow-Origin header for the whole domain.
    - Access-Control-Allow-Origin with wildcard origin
      - Note: You will not be able to use Access-Control-Allow-Credentials
      - You must add Origin to the Vary header
-

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Réseaux de champions

- Qui est responsable d'ajouter la « security story » dans le sprint pendant la rencontre de planification?
  - Le champion de la sécurité!

- Qu'est-ce qu'un champion de la sécurité?

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Réseaux de champions

- Avantages d'un réseau de champions de sécurité applicative :
  - Sécurité faite plus tôt dans le cycle de développement
  - Point de contact en cas de problème de sécurité
  - La sécurité « scale » plus efficacement qu'avoir une grosse équipe de sécurité AppSec centrale
    - Irréaliste de penser que l'équipe centrale assiste à toutes les rencontres de sprint
  - Meilleur arrimage avec la réalité de l'équipe
    - Le champion de sécurité connaît le vocabulaire de la sécurité mais est aussi un expert dans son projet
  - Meilleure promotion de la culture de sécurité

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Réseaux de champions

- Un réseau bien développé augmente la maturité de l'organisation
  - Plein de gens motivés en sécurité
  - Échanges de connaissances
  - Discussions stimulantes

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Équipe AppSec

- Réseau de champions ne rend pas une équipe AppSec centrale inutile
  - L'équipe centrale est une équipe d'experts qui soutiennent et orientent le réseau de champions
- Souvent les organisations matures ont des équipes de sécurité IT et de sécurité applicative distinctes
  - L'emphase n'est pas nécessairement la même, p. ex. :
    - Sécurité IT « infra » : ne pas mettre de *post-it* avec votre mot de passe sur votre bureau
    - Sécurité AppSec : il faut filtrer les entrées et encoder les sorties

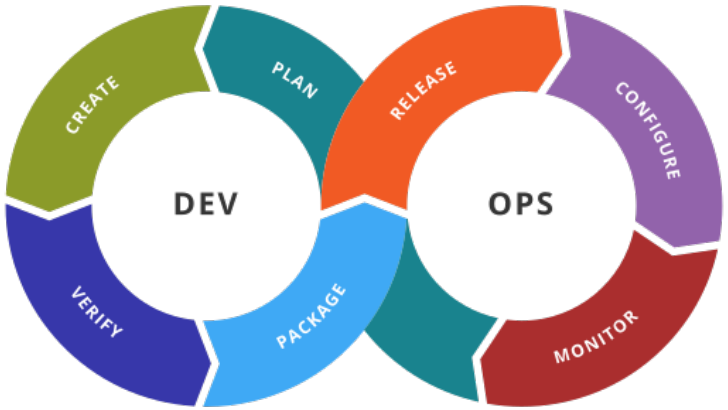
 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

DevOps

- Prochain niveau de maturité pour beaucoup d'organisations
  - Une étape de maturité difficile pour la sécurité
  - Tout va plus vite, souvent la sécurité « lag » et est manuelle
- Déploiement plusieurs fois par jour
  - Pas le temps de faire une évaluation de risque d'un mois… 50 fois par jour
  - Par le temps de faire un « pentest » d'une semaine… 50 fois par jour
- « Shift-Left, Shift-Right, Automate »

Security story



By Kharnagy - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=51215412

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Dette technique en DevSecOps

- La dette technique n’est jamais souhaitable
  - Dans un contexte DevOps, ça devient invivable
- Compenser avec un processus manuel n’est plus possible
  - Exemple CSRF
- Accélérer en faisant encore juste des scans/pentests à la dernière minute ne marche pas
  - Développeurs débordés –> tourner les coins ronds -> création de dette technique

 © 2017 Bentley Systems, Incorporated

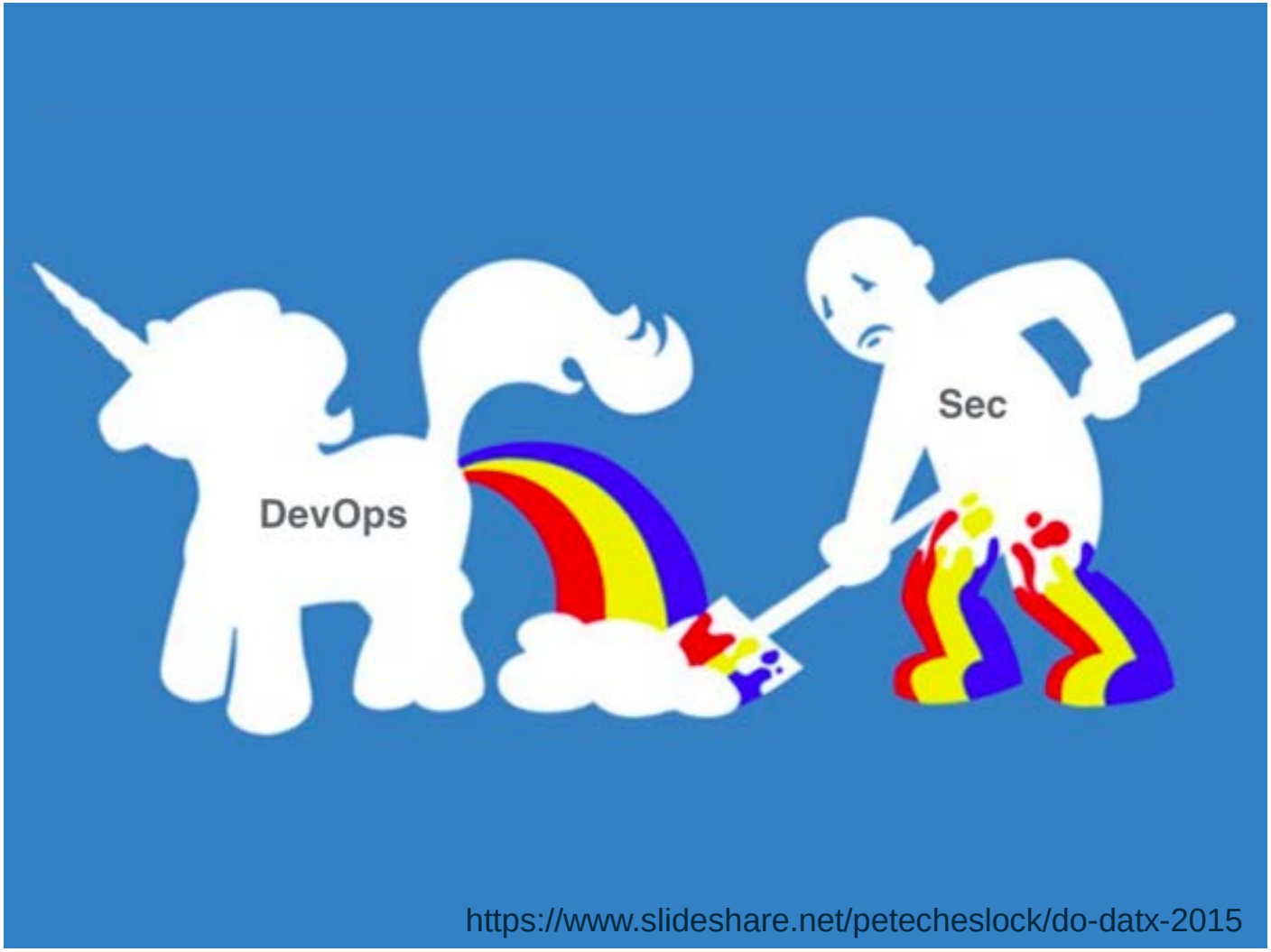
Bentley AppSec Team

Dette technique en DevSecOps

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Éviter ça



 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

DevOps

- OWASP SAMM ne donne pas plus de points de maturité même si tout est automatisé
  - Microsoft SDL a un document DevOps depuis 2015 déjà (pas super)

- DevSecOps
  - Automatisation
  - Accepter certains risques
    - On sait qu'on est capable de faire un « revert » en quelques minutes ou de faire une correction
    - On sait que n'importe quel problème a un impact minimal car tout est bien ségrégué, mitigé et qu'on a une défense en profondeur

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Conclusion

 © 2017 Bentley Systems, Incorporated

Conclusion

- Les modèles de maturité tel OWASP SAMM peuvent vous aider à prioriser vos efforts d'amélioration en sécurité
  - Un support crédible pour orienter et justifier vos décisions
- Il existe d'autres modèles comme Microsoft SDL qui peuvent être aussi utiles en fonction de votre contexte
- Les modèles ne sont pas parfaits et il faut considérer d'autres mesures dans des contextes Agile et DevOps :
  - automatisation
  - « security stories »
  - réseau de champions

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Références

- OWASP SAMM 1.5
  - https://www.owasp.org/index.php/OWASP_SAMM_Project
- OWASP SAMM 2.0 Draft
  - https://github.com/OWASP/samm/tree/master/v2.0
- Microsoft SDL
  - https://www.microsoft.com/en-us/sdl/
- Microsoft TMT
  - https://www.microsoft.com/en-us/download/details.aspx?id=49168
- SAFECore Agile Stories
  - Http://safecode.org/publication/SAFECode_Agile_Dev_Security0712.pdf

 © 2017 Bentley Systems, Incorporated

Contact

- Louis.Nadeau@Bentley.com
- Louis.Nadeau@Owasp.org

 © 2017 Bentley Systems, Incorporated

Bentley AppSec Team

Questions?

 © 2017 Bentley Systems, Incorporated

Bentley® AppSec Team