Résumer de lecture :

1. How much architecture? Reducing the up-front effort
   1. L’utilisation d’architecture template est utile dans le développement agile car elle permet de ne pas faire faire beaucoup de travail up-front et de retenir les bienfaits d’une architecture définit explicitement et il est plus facile de les modifier en cour de route ce qui réduit le risque de planifier l’architecture dès le départ.  
        
      L’expérience des architecture est aussi importante puisqu’elle fait en sorte qu’on a moins besoins de la documenter et que les erreurs sont moins fréquente dû à la connaissance de l’architecture par l’équipe.
2. How much up-front? A grounded theory of agile architecture
   1. Un danger de ne pas faire assez d’architecture up-front est qu’une architecture accidentelle émerge et que plus tard l’équipe passe plus de temps à réparer l’architecture qu’à livrer quelque chose qui est important aux yeux du client.  
        
      Trop planifier l’architecture retarde au mieux le projet, mais peut mener à beaucoup de travail de refactoring si les exigences du projet changent significativement.
   2. Le texte donne six menaces au développement qui sont adresser par la méthode agile. Ex : Afin de répondre à la menace du changement des exigences, il faut répondre au changement et s’adapter.  
        
      Six forces qui influe sur le travail sont répondue par 5 stratégies.
   3. To maximise agility, a team must find an appropriate tradeoff between a full up-front architecture design and a totally emergent design.
3. Agile software development: The business of innovation
   1. Creativity, not voluminous written rules, is the only way to manage complex software development problems and diverse situations.
   2. Le document défini plein de pratique de développement agile. Ne parle pas particulièrement d’architecture, mais donne une idée du niveau général de préparation qu’une équipe agile devrait avoir.
   3. Le principal est d’agir sur les changements et de suivre ces changements et non de les combattre.
4. Software architecture and agile software development: a clash of two cultures?
   1. Les praticiens de l’agile perçoivent et décrivent l’architecture logiciel comme étant tout le mal dans se monde et voient l’implémentation de frature YAGNI (you ain’t gonna need it)
   2. La définition de l’architecture et le concept d’architecture est souvent flou. Ce ne sont pas tous les designs qui sont des architectures.
   3. Donne 7 dimensions qui sont souvent conflictuel entre l’architecture et la méthode de développement agile.
5. Software architecture-centric methods and agile development
   1. Les activités d’architecture peuvent complémenter les activités d’un développement agile.

Bass, L., Clements, P., and Kazman, R. 2003. Software Architecture in Practice. 2nd ed. Addison-Wesley.

[6] ISO/IEC 42010 CD1, 2010. Systems and software engineering—Architectural description, committee draft #1. (February 2010) ISO.