**1. Bibliographic data**

* **Title:** Requirements in the 21st Century: Current Practice & Emerging Trends.
* **Authors:** Sean Hansen, Kalle Lyytinen, Nicholas Berente
* **Year:** 2009
* **Publication:** January

**2. Theme of the paper**

**Scientific Area:** Requirements Engineering and Information Systems Design.

**Specific Topics:**

* Trends and techniques in requirements management in the current times
* The existent gap between theory and practice in requirements engineering
* New challenges in designing integrated and complex systems
* The effects of software integration and system transparency on user perception

**3. Synthesis of the Paper**

**Motivation and Importance of Research**

The study’s target is to assess the current state of requirements engineering practices and try to identify some differences between theory thought in an academic environment and the actual industrial practices that are being used currently. It also tries to understand how integration demands and evolving and more complex systems are changing the more like traditional approaches to requirements collecting and defining.

**Main Points of the Background Information and State of the Art**

Traditional approaches tend to have more focus on requirement discovery, specification, and verification as distinct processes. However, practically, the organizations do not clearly distinguish these phases, treating them as a continuous process. Sometimes some of those phases are grouped up and used at the same time. Formal models and standardization are applied inconsistently and adaptively, depending on the specific challenges of each project. These inconsistencies can happen even inside of the same organization, with the same teams.

**Main Findings, Results and Novelty**

Process Focus: requirement processes are increasingly being driven by business process needs rather than by isolated software development. The process is getting a lot more flexible as time goes on.

* **System Transparency:** Users demand seamless experiences across applications. That means that the users don’t want to feel changes into the software while using different devices or platforms.
* **Integration Over Development:** Efforts need to be more focused on integrating existing applications rather than creating entirely new ones from scratch. Organizations need to increasingly rely more on modular softwarerather than isolated applications.
* **Distributed Requirements:** Requirements gathering are increasingly distributed across multiple teams, organizations, and geographical locations.
* **Layered Requirements:** Iterative requirement development across multiple levels of abstraction, design focus, or timing. This type of approach supports agile development, which is the type of development that takes user feedback to better develop software, which is what is needed into business process currently.
* **Packaged Software Adoption (COTS):** A shift toward vendor-driven requirements due to the increased use of commercial off-shelf software. Organizations need to try to adapt their workflows to the software rather than trying to customize it fully.
* **Architectural Centrality:** Architectural requirements play a big role, influencing product and application-level requirements.
* **Interdependent Complexity:** While some aspects of complexity have been reduced, overall system complexity has significantly increased.
* **Design Fluidity:** Requirements processes must accommodate continuous system evolution even after deployment.

**Main Conclusions and Discussion Points**

* It’s possible to see the evolution that was made by the requirements engineers. Now instead of just documenting the needs of the clients, now they are trying more to find solutions by using different trends like making mockups.
* There is a clear separation between the theorical approach into requirement phases described in the paper (discovery, specification, validation) and what is done by the companies. Companies now have a much more approach towards being iterative instead of following the 3-phase approach.
* The document suggests new directions for research, including better ways of dealing with highly complex and constantly evolving systems.

**4. Questions and Reflection**

**Questions Raised by the Reading of the Paper**

* Are engineering requirements capable of adapting to the increasing complexity and bigger projects (like the ones that tend to have global impact) of modern systems?
* How can system transparency and integration be balanced with the need for security and user privacy?
* How can requirement models remain flexible enough to accommodate continuous system evolution and what needs to change in the approach with the clients in those situations?

**Opinion about the paper**

The paper is a good and realistic description of challenges in requirements engineering in the current times. The paper correctly talks about the gap between academic studies and actual practices that are used in the actual working industry. The paper would be better if it described, at least, some practical approaches to dealing with the challenges it raises. Those didn’t need to be explored deeply but, at least, try to expose some ideas at high level.

**What to retain for my future research/professional practice**

**• Business process orientation:** Requirements should be business process driven, not technical requirements

**• System transparency:** Smooth user experiences (UI/UX) are a new concern

**• Integration over development:** Integration and working between systems is more important than developing in isolation

**• Iterative and flexible methods:** Requirement engineering needs to be attuned to systems evolving day in and day out

**• Balancing formality and adaptability:** There is a need to strike a balance between formal requirements processes and dynamic