
Méthodes de recherche empirique en ingénierie des SI

Principes et applications

Saïd Assar

Télécom École de Management, Institut Mines-Télécom
9 rue C. Fourier, F-91011 Evry, France
said.assar@telecom-em.eu

RÉSUMÉ. Cet article est une introduction synthétique aux méthodes empiriques de recherche en ingénierie informatique. Ces méthodes font partie intégrante des sciences de la nature et des sciences du comportement. Dans l'ingénierie des SI en tant que science de conception, ces méthodes sont essentielles pour évaluer et valider l'artéfact conçu, qu'il soit un algorithme, un logiciel, une méthode ou un outil d'ingénierie. Le propos de cet article est de présenter les principaux concepts et termes indispensables pour comprendre ces méthodes, et de mettre en évidence leur capacité à produire une connaissance nouvelle indispensable pour le concepteur d'un artéfact. Cette présentation didactique est illustrée par l'analyse de plusieurs études empiriques publiées récemment. L'analyse met en évidence la multiplicité des approches possibles, le potentiel de certaines méthodes spécifiques issues des sciences sociales, ainsi que l'importance de la réplication en tant que moyen pour confirmer un résultat et contribuer à sa validation sur une plus large échelle.

ABSTRACT. This paper is a synthetic introduction to empirical research methods in computer science engineering. These methods are fully integrated into natural and behavioral sciences. In IS engineering as a design science, these methods are important to evaluate and validate designed artifacts, whatever the artifact is an algorithm, a piece of software, a method or an engineering tool. The topic of this paper is to present the main concepts and terms necessary to understand these methods and to highlight their capacity in producing new knowledge that is essential for the artifact designer. This didactic presentation is illustrated by analyzing a set of empirical studies published recently. The analysis highlights the multiplicity of research approaches, the potential of certain specific methods from social science, together the importance of replication as a mean to confirm a result and to validate it at a larger scale.

MOTS-CLÉS : méthodologie de recherche, recherche empirique, sciences de conception, design science, expérimentation, ingénierie des systèmes d'information, génie logiciel.

KEYWORDS: research methodologies, empirical research, design science, experimentation, information systems engineering, software engineering.

DOI:10.3166/ISI.20.6.11-33 © 2015 Lavoisier

1. Introduction

L'expérimentation est une étape capitale de la démarche scientifique. L'analyse d'un problème et la formulation d'une question de recherche amènent le chercheur à interroger le corpus des connaissances disponibles et à formuler des *hypothèses* en tant qu'explications possibles au problème étudié (Bunge, 1967, cf. figure 1). Ces hypothèses doivent être vérifiées dans leurs conséquences, et pour cela être soumises au contrôle de l'expérience. L'évaluation expérimentale des hypothèses requiert l'élaboration de techniques pour tester les conséquences attendues ; ceci nécessite la définition de *mesures* pour récolter données, traces et observations empiriques dont l'analyse va confirmer ou infirmer la véracité des hypothèses émises et permettre ainsi l'enrichissement du corpus des connaissances par des réponses, partielles ou complètes, au problème initial.

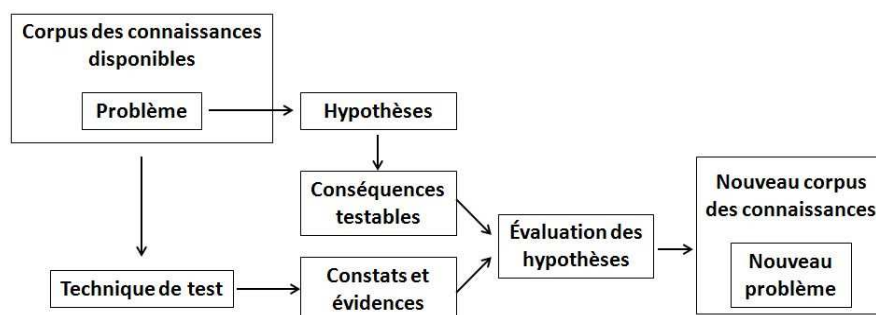


Figure 1. Le cycle fondamental de la recherche scientifique (Bunge, 1967, p. 9)

Ce cycle de la recherche scientifique est considéré comme universel dans les sciences de la nature. Il correspond à une vision rationaliste et *positiviste* de la démarche scientifique, où l'on postule que les liens de causalité existent indépendamment des acteurs et que le chercheur peut les découvrir. Néanmoins, il ne s'agit pas de la seule manière, scientifiquement reconnue, d'acquérir des connaissances et d'appréhender la réalité. Les sciences du comportement, en particulier la psychologie et la sociologie, ont développé des approches nouvelles pour comprendre et expliquer des phénomènes dans lesquels la dimension humaine est prépondérante. Ces approches reposent sur l'observation des acteurs humains impliqués dans le phénomène étudié et sur *l'interprétation* des données empiriques récoltées. Ces données souvent subjectives portent généralement sur des aspects spécifiquement humains tels que la perception, la compréhension ou les convictions de ces acteurs (Sofaer, 1999). Ces approches se réclament d'une vision *constructiviste* de la démarche scientifique, elles postulent que la connaissance est construite par l'acteur humain et qu'elle est sujette à l'interprétation du chercheur.

Qu'en est-il de l'ingénierie des systèmes d'information (SI) ? Historiquement, l'informatique est enracinée dans les mathématiques et les sciences formelles.

Néanmoins, le génie logiciel et l'ingénierie des SI sont des sciences de conception, des sciences de *l'artificiel*, où l'objectif n'est point de comprendre le monde tel qu'il est, mais d'imaginer comment il *pourrait être* et de spécifier des artefacts qui satisfont les besoins de la vie humaine (Simon, 1969). Dans les *sciences de conception*, un enjeu majeur est *l'adéquation* entre l'artefact et le contexte dans lequel il sera utilisé, ou tout au moins, celui pour lequel il a été conçu.

Le cadre théorique des sciences de conception (*Design Science*) de Hevner *et al.* (2004) a fortement contribué à clarifier le rattachement de l'ingénierie des SI aux sciences de conception. Dans ce cadre (cf. figure 2), trois cycles d'investigation sont à l'œuvre dans tout projet de recherche en sciences de conception. Le premier correspond à l'utilisation des connaissances et des technologies disponibles pour construire un artefact afin de résoudre des problèmes et répondre à des besoins. Le deuxième cycle s'attache à relier de manière concrète et pertinente ces problèmes et ces besoins à un environnement organisationnel et humain spécifique. L'artefact devra par ailleurs être confronté à la réalité tangible de ces besoins et de ces problèmes. De ces deux cycles émergent des connaissances nouvelles concernant d'une part, l'artefact, son adéquation avec l'environnement cible et sa capacité à répondre aux besoins, et d'autre part, ce qui a été appris de la conception et de l'évaluation de celui-ci, et des éventuelles critiques, extensions et additions aux connaissances théoriques mobilisées à cet effet. Le troisième cycle (cycle 3) considère la formulation rigoureuse de cette connaissance nouvelle, son potentiel de généralisation (à d'autres artefacts ou d'autres contextes) et son insertion dans le corpus des connaissances.

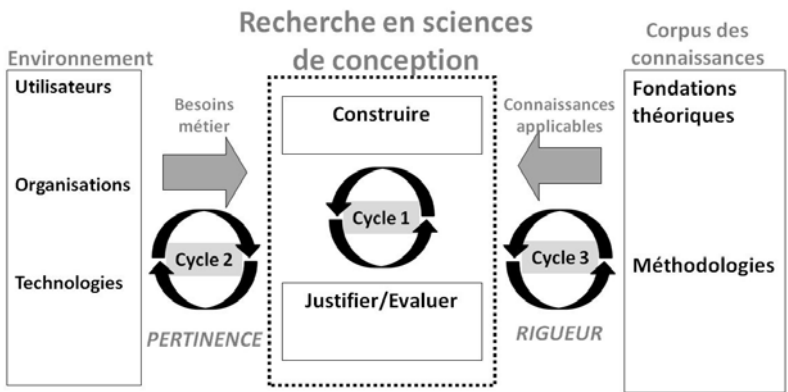


Figure 2. Trois cycles dans la recherche en sciences de conception (Hevner et Chatterjee, 2010a, p. 16)

Le cadre de Hevner *et al.* (2004) met en exergue l'importance de l'évaluation dans la recherche en sciences de conception. Dans les cycles 1 et 2, on doit inspecter l'artefact, vérifier qu'il respecte le cahier des charges, et jauger la satisfaction des

besoins et l'adéquation entre l'artéfact et son contexte d'usage. L'évaluation joue aussi un rôle important dans le cycle 3 où toute connaissance nouvelle, telle que l'applicabilité d'une méthode ou d'une théorie, doit être rigoureusement évaluée. Dans les trois cas, l'évaluation repose sur la récolte et le traitement de données empiriques. Pour aboutir à des résultats scientifiquement valides, l'analyse de ces données doit suivre avec rigueur une démarche épistémologiquement fondée.

L'objectif de cet article est de présenter les méthodes empiriques de recherche et d'illustrer leurs applications à travers plusieurs exemples issus de la littérature en génie logiciel et en ingénierie des SI. Il constitue une brève synthèse sur un sujet complexe avec des pointeurs vers des publications plus détaillées. Notre propos est de démontrer, exemples à l'appui, leur potentiel de production de connaissances et leur importance pour la recherche en ingénierie des SI. En effet, plusieurs études laissent penser que ces méthodes ne sont pas très répandues au sein de la communauté des chercheurs en informatique. En 1995, une analyse de 403 publications sur la période 1991-1994 a montré que 40 à 50 % des articles nécessitant une évaluation n'en comportaient pas (Tichy *et al.*, 1995). Un constat similaire est fait dans une étude portant sur 622 articles publiés entre 1985 et 1995 (Zelkowitz et Wallace, 1997). Enfin, depuis l'article fondateur de Basili *et al.* (1986) sur l'expérimentation en génie logiciel, des appels récurrents sont lancés pour introduire plus d'empirisme dans la recherche en informatique. On peut citer l'appel de Kitchenham *et al.* (2004) pour un génie logiciel basé sur les constats empiriques, ainsi que l'analyse de Denning (2013) pour ancrer l'approche expérimentale dans les fondements scientifiques de l'informatique.

Quel serait l'apport de l'empirisme en informatique et en ingénierie des SI ? D'après (Tichy, 1998), deux avantages majeurs sont à considérer. Tout d'abord, et tel suggéré par le cadre des sciences de conception (cf. figure 2), l'évaluation empirique des artéfacts ou des hypothèses et suppositions sous-jacentes, aboutit à une connaissance nouvelle concernant l'adéquation, la pertinence ou la performance de ces artéfacts. Ce qui permet d'éliminer les voies de recherche stériles, les approches erronées ou les fausses hypothèses. Ensuite, l'observation et l'analyse empirique des pratiques autour de l'artéfact peuvent fournir de nouveaux éléments de compréhension, parfois inattendus ou contradictoires. A titre d'exemple, dans une étude de cas en industrie où un même logiciel avait été développé avec un langage classique (C) et ensuite avec un langage orienté objet (C++), Hatton (1998) remet en cause les avantages supposés de l'orienté objet en observant que les temps de résolution des erreurs sont largement supérieurs pour le logiciel écrit en C++, ce qui laisse supposer que l'orientation objet alourdit la maintenance des logiciels.

Le reste de cet article est organisé comme suit. Dans la section 2, nous présentons les principaux termes et concepts indispensables pour appréhender les méthodes empiriques. Dans la section 3, en nous appuyant sur les principales publications dans le domaine, nous dressons un panorama des méthodes de recherche empirique avec une présentation brève de chacune d'elles. Afin d'illustrer notre propos, nous analysons dans la section 4 un échantillon de neuf études empiriques publiées dans la littérature. Cette sélection a été effectuée de manière à,

d'une part, traiter de problématiques pertinentes pour la communauté de recherche en ingénierie des SI, et d'autre part, illustrer la variété des méthodes empiriques applicables. Cet article se termine par une conclusion générale dans laquelle nous évoquerons brièvement le futur des recherches empiriques en informatique.

2. Terminologie et principaux concepts

Les méthodes empiriques de recherche font appel à un large éventail de termes et de concepts spécifiques. Elles reposent sur des raisonnements issus de paradigmes différents et qui permettent d'inférer des connaissances de natures différentes. Nous présentons brièvement ici les éléments qui nous semblent primordiaux pour en comprendre les fondements.

2.1. Mesure, variable et évaluation

La *mesure* détermine la valeur d'un attribut ou d'une propriété d'un objet ou d'un phénomène. Elle est réalisée à l'aide d'un *instrument* et nécessite la définition d'une *métrique*. Pour les artefacts en ingénierie informatique, les attributs et propriétés que l'on cherche à mesurer peuvent être concrets, tels que le nombre d'entités dans un schéma de base de données ou le temps de réponse d'un logiciel ; ou conceptuels et abstraits, par exemple la lisibilité du schéma ou la facilité d'utilisation du logiciel perçus par un utilisateur. En génie logiciel, Basili *et al.* (1994) ont introduit et développé une théorie importante de la mesure autour du principe GQM (*Goal/Question/Metric*).

Une mesure correspond généralement à une *variable* que le chercheur considère comme susceptible de caractériser l'objet étudié ou d'influer le phénomène observé. Un grand nombre de variables sont définies dans les sciences de conception : Prat *et al.* (2014) ont recensé et établi une hiérarchie de 28 variables (appelées *critères*). Une variable peut *dépendre* d'autres variables. Ces liens de dépendance, de corrélation ou de cause à effet entre variables définissent ainsi un *modèle empirique d'investigation* dans lequel la *variable dépendante* correspond à la finalité de l'étude. C'est le cas par exemple dans le célèbre modèle TAM (*Technology Acceptance Model*) où l'acceptation d'un logiciel est définie comme dépendante de l'utilité perçue et de la facilité d'usage (Davis, 1989).

L'*évaluation* se définit ainsi comme la réalisation d'un ensemble de mesures, relatives à un artefact ou à un phénomène qui s'y rattache, à l'aide d'un modèle empirique d'investigation. Le rôle et la place du modèle empirique définissent deux postures d'évaluation radicalement différentes : le modèle peut être posé comme une hypothèse à vérifier à l'aide d'une série de mesures, ou au contraire, émerger à partir d'un ensemble d'observations minutieuses. C'est l'objet de la prochaine section.

2.2. Expérimental versus empirique

L'expérience scientifique est une forme particulière d'empirisme. C'est une mise à l'épreuve, rigoureuse et systématique, de quelque chose pour en vérifier les propriétés (Larousse, 2015). L'expérimentation relève bien de l'empirisme car l'observation y joue un rôle fondamental. C'est une méthode d'évaluation où le chercheur dispose d'un certain contrôle sur l'environnement de l'évaluation (lieu, acteurs, contexte, etc.). Avec ce contrôle, et par le jeu des hypothèses et du modèle d'investigation qu'il souhaite tester, le chercheur espère mieux comprendre le phénomène étudié et découvrir des relations entre les variables de son modèle, et ainsi, émettre – ou valider – une connaissance théorique nouvelle. L'essor de l'approche expérimentale est étroitement lié au développement de la science à travers l'histoire moderne (Lecourt, 2012). La figure 3 dresse le cadre formel d'une expérimentation : le modèle d'investigation relie les variables et exprime les hypothèses du chercheur. Le chercheur n'ayant pas le contrôle sur l'intégralité de l'environnement, des facteurs confondants pourraient influencer sur les observations et remettre en cause la *validité interne* de l'expérience. L'identification de ces facteurs et l'analyse de leur effet potentiel est un élément clé de toute expérimentation.

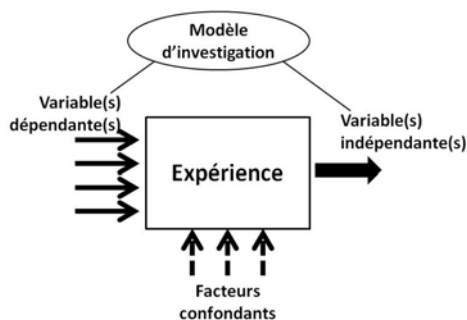


Figure 3. Synopsis d'une expérimentation (basé sur Wohlin et al., 2003, p. 14)

Néanmoins, l'empirisme en tant qu'approche pour produire des connaissances, ne se limite pas à l'expérimentation. L'empirisme est une posture philosophique générale qui s'attache exclusivement à l'observation et au classement des données sans l'intervention d'un système ou d'une théorie *a priori* (Larousse, 2015). Elle postule que l'on peut, à partir d'une accumulation d'observations, inférer des règles générales par un raisonnement inductif allant du concret observé et mesuré vers un abstrait conceptualisé et théorisé. C'est le cas avec les méthodes d'évaluation basées sur l'interprétation des observations empiriques. Ces méthodes relèvent du *constructivisme* et considèrent que la connaissance sur les choses est construite par l'acteur humain selon ses perceptions propres, sa culture, ses expériences passées, etc. Le chercheur interprète les constats empiriques récoltés pour en tirer une connaissance nouvelle. Cette connaissance est *a priori* descriptive et analytique, elle

peut apporter de précieux renseignements sur le phénomène étudié et sur les facteurs susceptibles de l'influer. Elle s'avère dans ce cas très utile pour explorer un phénomène nouveau ou peu connu (cf. §2.4). Néanmoins, le chercheur peut abstraire et conceptualiser les constats observés et élaborer ainsi un modèle empirique. Ce modèle, obtenu par un raisonnement inductif, peut par la suite devenir une hypothèse qui sera évaluée expérimentalement. Dans cette démarche *hypothético-déductive* d'investigation, le chercheur applique deux mécanismes de raisonnement logique : l'abduction et la déduction.

2.3. Déduction, induction et abduction

Pour que l'empirisme puisse produire une connaissance, il est nécessaire d'appliquer des *mécanismes d'inférence*. Trois formes d'inférence sont possibles, nous les présentons brièvement en nous basant sur l'exposé de David (1998).

La forme la plus simple est la *déduction*, c'est-à-dire l'application « mécanique » d'une règle sur la situation en cours, et qui produit nécessairement une conséquence prédéfinie :

Règle + Situation => Conséquence

Par exemple, la règle « *l'implication de l'utilisateur final tout le long d'un projet informatique garantit le succès de celui-ci* » permet de déduire logiquement que le succès sera au rendez-vous si l'utilisateur final est effectivement impliqué. La remise en cause d'une règle se fait généralement lorsqu'un contre-exemple est observé, dans le monde réel ou dans une expérience.

L'induction est un raisonnement différent qui permet d'inférer une règle :

Situation + Conséquence => Règle

En reprenant le même exemple, si le chercheur observe que les projets informatiques dans lesquels l'utilisateur final est impliqué se terminent toujours avec succès, il peut alors induire une règle de corrélation – ou de cause à effet – entre l'implication de l'utilisateur final et le succès d'un projet. Cette règle empirique peut être erronée : le succès peut être dû en réalité à d'autres facteurs (confondants) tels que l'expertise de l'équipe projet ou la simplicité des projets observés. Pour valider cette règle, ou à défaut, augmenter sa validité, le chercheur devra accumuler d'autres observations similaires pour corroborer sa règle, ou mener des expérimentations rigoureuses pour démontrer que celle-ci s'applique dans le cas général (ou à défaut déterminer les conditions d'applicabilité de la règle).

L'abduction est une forme particulière d'induction, appelée aussi « *inférence de la meilleure explication* » (Douven, 2011). Devant un faisceau d'observations, le chercheur fait l'hypothèse qu'une règle déjà connue – ou inventée pour l'occasion – explique la situation en cours :

Règle + Conséquence => Situation

En reprenant l'exemple précédent, si le chercheur observe qu'un ensemble de projets informatiques se terminent avec succès, et qu'une théorie existante affirme que les projets dans lesquels l'utilisateur final est impliqué se terminent avec succès, il va alors faire l'hypothèse que ce succès est dû à l'implication de l'utilisateur final. Le chercheur a fait cette hypothèse car elle lui semble la plus *plausible* étant donné le corpus de connaissance dont il dispose. Il aurait pu bien sûr faire d'autres hypothèses, à tort ou à raison. Comme pour l'induction directe, l'hypothèse émise peut être erronée si le succès dans les projets observés est en réalité dû à l'expertise de l'équipe projet. L'abduction est ainsi une *conjoncture*, une forme de *pari spéculatif* sur une explication possible au phénomène observé. Elle fait appel à la créativité du chercheur pour proposer une explication nouvelle, originale ou innovante au phénomène étudié. Cette explication devra bien sûr être validée par d'autres observations, ou à l'aide d'expérimentations rigoureuses pour démontrer que c'est bien l'explication du chercheur qui s'applique et non pas une autre, et également pour délimiter le périmètre de son applicabilité.

Ces trois formes d'inférence interviennent à différents moments dans la recherche empirique. Elles composent un cycle récursif *abduction* → *déduction* → *induction* (David, 1998, p. 5). Ce cycle est mis en œuvre à travers une panoplie de méthodes de recherches pour *comprendre* et *explorer* un phénomène, et par la suite, élaborer une *explication* rigoureusement corroborée par l'observation.

2.4. Explorer versus expliquer

A la lumière de ce qui vient d'être présenté, il devient possible d'apporter une clarification de principe concernant la distinction entre une recherche empirique *exploratoire* et celle qui est *explicative*. Cette distinction est liée avant tout à la nature de la connaissance inférée et aux raisonnements sous-jacents. Une recherche *explicative* implique l'inférence et l'évaluation d'une règle empirique, ou la proposition d'une hypothèse – ou théorie – que vient confirmer une expérimentation ou une série minutieuse d'observations (Wohlin *et al.*, 2012, p. 9). Cette proposition est alors candidate à intégrer le corpus de connaissance selon le cycle 3 de la recherche (cf. figure 2). Par opposition, une recherche exploratoire reste à un stade de description des observations récoltées, d'analyse et de compréhension du problème, et d'identification des variables et des relations qui sont en jeu. Elle constitue néanmoins une *théorisation* indispensable qui prépare le terrain à une investigation plus poussée, en particulier lorsque le problème étudié est nouveau ou méconnu (Weick, 1995). Cette distinction n'est pas totalement binaire, mais constitue un continuum entre deux formes complémentaires de recherche empirique.

2.5. La question de la répliquabilité

Considéré de manière isolée, le résultat d'une série d'observations empiriques ou d'une expérimentation peut difficilement être à lui seul représentatif de la réalité

sous-jacente. La vérification minutieuse des résultats précédemment obtenus est essentielle pour garantir que ce qui a été constaté se produit de manière régulière et stable, qu'il n'est pas dû au hasard ou à une conjonction de facteurs autres que ceux identifiés. La réplication est la répétition d'une recherche empirique pour revérifier ses résultats et augmenter ainsi la confiance qu'on y accorde (Shull *et al.*, 2008). Elle implique la reproduction de l'étude empirique dans des conditions identiques ou similaires. L'interprétation du résultat de la réplication dépendra du niveau de similarité entre l'étude initiale et la réplication.

En effet, le niveau de similarité entre une réplication et l'étude empirique initiale est une question essentielle. Peut-elle être réalisée par les mêmes chercheurs ou doit-elle être réalisée par d'autres ? Doit-on utiliser les mêmes données, le même procédé opératoire, les mêmes outils d'analyse, etc. ? Ces interrogations, initialement posées dans les sciences de comportement (Rosenthal, 1991), ont fait l'objet d'une analyse intensive dans le domaine du génie logiciel empirique. L'objectif est d'harmoniser la terminologie et les démarches de réplication. Au-delà du constat qu'une réplication exacte d'une étude empirique – *une répétition à l'identique* – est pratiquement impossible (Brooks *et al.*, 2008, p. 370), une telle réplication n'aurait en réalité qu'un intérêt limité. Certes, son succès confirmerait les conclusions et la validité interne de l'étude initiale, mais elle n'apporterait aucun élément nouveau concernant la généralisation de ses conclusions (Kitchenham, 2008). En réalité, ce sont les différences entre l'étude initiale et la réplication qui ont le potentiel de générer de nouvelles connaissances (Juristo et Vegas, 2009). En cas de succès, une réplication différenciée confirmerait nettement les conclusions de l'étude initiale et étendrait le périmètre de validité de ses résultats. En cas d'échec, cela ne remettrait pas entièrement en cause l'étude initiale, mais constituerait un sérieux indicateur de l'existence de facteurs inconnus jusqu'alors, qui ont influencé le résultat de l'étude et qui contribuent donc à en limiter le périmètre de validité (Rosenthal, 1991, p. 6).

Comment caractériser le niveau de similarité d'une réplication ? Par définition, une réplication est une étude empirique ayant des éléments en commun avec l'étude initiale (Juristo et Gómez, 2012). Au-delà du terme « réplication exacte », les termes tels que « réplication non exacte », « réplication conceptuelle » ou « réplication indépendante » sont utilisés pour désigner le cas où un – ou plusieurs – paramètre(s) de l'étude initiale (acteurs, contexte, etc.) sont changés. Juristo et Vegas (2010) clarifient l'usage de cette terminologie dans le champ du génie logiciel expérimental à l'aide d'une classification synthétique des multiples définitions – toutes disciplines confondues – de la réplication. Ils considèrent ainsi l'utilisation ou non de *la même méthode expérimentale* comme le principal facteur discriminant. Cette classification est par la suite affinée dans (Gómez *et al.*, 2014) selon quatre paramètres qui peuvent être maintenus à l'identique ou changés : le protocole, le procédé opérationnel, la population et les expérimentateurs. Ceci définit une échelle de seize formes possibles de réplication, allant de la *répétition* où tous les paramètres sont identiques, à la *reproduction* où tous les paramètres sont modifiés.

3. Aperçu des méthodes empiriques en ISI et GL

Il y a une multitude de manières et d'approches différentes pour concevoir une recherche empirique. Ces méthodes se distinguent par l'objectif de recherche, les termes utilisés, le paradigme épistémologique sous-jacent, la nature des données récoltées, le niveau de contrôle du chercheur sur l'environnement de l'étude, etc. De plus, selon les disciplines, les termes ne sont pas utilisés de manière homogène et peuvent parfois désigner des méthodes substantiellement différentes.

Nous avons réuni dans le tableau 1 les principales méthodes de recherche empiriques mentionnées dans les publications de référence sur le sujet en ingénierie informatique. Ce tableau a été obtenu en confrontant la liste des douze méthodes indiquées dans (Zelkowitz et Wallace, 1998) avec celles présentées dans quatre sources plus récentes : (Wohlin *et al.*, 2003 ; Easterbrook *et al.*, 2008 ; Shull *et al.*, 2008 ; Wohlin *et al.*, 2012). Nous avons gardé le regroupement initial de Zelkowitz et Wallace (1998) selon trois catégories : observationnelle, historique et contrôlée. De plus, nous avons caractérisé chaque méthode avec trois propriétés :

- **La portée** peut être *exploratoire* ou *explicative* (cf. §2.4). Alors que toutes les méthodes peuvent être utilisées de manière exploratoire pour décrire un phénomène, le comprendre et l'analyser, il n'en est pas de même lorsqu'il s'agit d'élaborer une explication avec un lien de cause à effet ou de corrélation entre variables.

- **Le contexte** d'application de la méthode est *in vivo* si la méthode peut – ou doit – être utilisée dans un contexte réel d'occurrence du phénomène (entreprise ou organisation). Il est *in vitro* si la méthode a vocation à être utilisée dans un laboratoire ou un environnement expérimental. Plusieurs méthodes peuvent être utilisées indifféremment dans l'un ou l'autre.

- **L'approche** de la méthode de recherche est *quantitative* si elle repose sur l'analyse numérique des données recueillies, elle est *qualitative* si elle fait appel à une interprétation des données par le chercheur. Cette propriété fait indirectement référence au paradigme épistémologique sous-jacent (*positiviste* ou *constructiviste*)¹.

La première méthode est l'**assertion**. C'est une forme rudimentaire d'évaluation, le chercheur se contente de tester lui-même un artéfact sur un jeu d'essai et de constater le résultat. Le niveau de validation apporté est faible, l'usage de cette méthode doit être considéré comme une étape exploratoire qui prépare à des évaluations plus élaborées.

Dans l'**observation de projets** (*project monitoring*), le chercheur recueille et analyse des données disponibles pour un ensemble de projets en cours. Ces données

1. Nous attirons l'attention du lecteur sur le fait qu'une démarche positiviste ne se limite pas à l'exploitation de données quantitatives mais peut aussi s'élaborer avec des données de nature qualitative, ou même une combinaison des deux. Ces aspects dépassent néanmoins le cadre introductif de cet article, le lecteur intéressé peut consulter des références telles que (David, 1998) ou (Runeson *et al.*, 2012).

sont généralement enregistrées par les acteurs des projets étudiés, le chercheur n'ayant pas de contrôle sur le choix de ces données ni sur leur niveau de détail.

L'ethnographie est une méthode originaire des sciences sociales. Son objectif est l'étude descriptive et analytique sur le terrain des mœurs et pratiques d'une population. Elle vise la construction d'un récit crédible, rigoureux et authentique en donnant la parole aux acteurs dans leur contexte local. Cette méthode relève donc d'une approche qualitative où le chercheur interprète le discours des acteurs. Le recueil des données fait appel à la technique d'entretien et/ou à l'observation participante (Easterbrook *et al.*, 2008, p. 300). L'entretien est dans ces cas libre ou *semi-structuré* pour donner la parole aux acteurs et permettre l'émergence de nouvelles connaissances sur le phénomène étudié grâce à des questions ouvertes².

Tableau 1. Principales méthodes empiriques

Catégorie	Méthode	Portée		Contexte		Approche	
		Exploratoire	Explicative	<i>In vivo</i>	<i>In vitro</i>	Quantitative	Qualitative
Observationnelle	1. Assertion	x			x	x	x
	2. Observation de projets	x		x		x	x
	3. Ethnographie et entretiens semi-structurés	x	x	x	x		x
	4. Focus group	x	x	x	x		x
	5. Enquête	x	x	x	x	x	
	6. Étude de cas	x	x	x		x	x
	7. Étude longitudinale de terrain	x	x	x		x	x
	8. Recherche action	x	x	x		x	x
Historique	9. Revue de littérature	x	x	NA		x	x
	10. Leçons apprises	x		x	x	x	x
	11. Analyse <i>post-mortem</i>	x		x	x	x	x
	12. Fouille de données	x	x	x	x	x	
Contrôlée	13. Expérience(s) contrôlée(s)	x	x	x	x	x	x
	14. Analyse comparative	x	x		x	x	x
	15. Simulation	x	x		x	x	x

2. Inversement, une *interview structurée* est une technique d'enquête en face à face où l'on utilise un questionnaire avec questions fermées et réponses prédéfinies.

Le **Focus Group** est une technique de recueil de données qualitatives plus élaborée que l'entretien semi-structuré ; c'est aussi une méthode de recherche à part entière étant donnée l'organisation spécifique qu'elle requiert pour sa mise en œuvre. Les entretiens sont organisés par petits groupes et le chercheur va exploiter l'interaction et le débat contradictoire entre les membres du groupe pour faire émerger de nouvelles idées et connaissances (Kitzinger, 1995). Le déroulement du Focus Group peut être facilité par un logiciel de travail collaboratif et permettre ainsi une interaction plus riche et plus sophistiquée (Kontio *et al.*, 2008).

L'enquête (*survey*) est une méthode quantitative basée sur l'administration d'un questionnaire à un échantillon représentatif d'une population. Elle fait appel à l'analyse statistique pour élaborer des résultats et tirer des conclusions. Même si elle peut être utilisée pour explorer un phénomène et alimenter la réflexion du chercheur, elle est surtout mise en œuvre pour tester des hypothèses et produire une explication. Son pouvoir explicatif réside dans la diversité des tests statistiques qui peuvent être appliqués. C'est une méthode prééminente en recherche empirique avec plusieurs textes de référence sur le sujet, tels que celui de Pfleeger et Kitchenham (2001).

Issue aussi des sciences sociales, **l'étude de cas** (*case study*) est une autre méthode prééminente en recherche empirique par sa disposition à observer un phénomène dans son contexte d'occurrence. De multiples formes de données, aussi bien quantitatives que qualitatives, sont récoltées durant l'étude, et contrairement à la simple observation de multiples projets, la récolte des données est dirigée par l'objectif de recherche. C'est une méthode qui est parfois confondue par les chercheurs en ingénierie avec l'assertion, c'est-à-dire le test par le chercheur lui-même d'un artefact sur un exemple ou un « cas » (Easterbrook *et al.*, 2008, p. 296). Une étude de cas peut être exploratoire pour découvrir un problème, délimiter son périmètre et identifier les variables en jeu. Elle peut aussi être confirmatoire, pour tester les explications produites par une théorie existante, et en particulier, pour établir les limites d'applicabilité de cette théorie, ou pour rapporter un contre-exemple susceptible de remettre en cause cette théorie (Runeson *et al.*, 2012).

L'étude longitudinale de terrain (*field study*) est une forme particulière d'étude de cas où un phénomène est observé à de multiples reprises sur une échelle étendue de temps. Le principe est de suivre le déroulement d'un phénomène à travers le temps pour analyser son évolution, cerner les facteurs qui y contribuent et y apporter une nouvelle compréhension (McLeod *et al.*, 2011). Elle est adaptée à l'étude de phénomènes complexes qui s'étalent sur de longues périodes temporelles.

La **recherche action** est une autre méthode d'investigation issue des sciences sociales. Quoique peu d'études utilisent cette méthode (Santos et Travassos, 2009), elle suscite actuellement un regain d'intérêt en tant que méthode particulièrement pertinente pour les sciences de conception (Sein *et al.*, 2011). En effet, dans la recherche action, le chercheur est lui-même acteur du phénomène qu'il veut observer, et l'action qu'il est susceptible d'apporter peut donc contribuer à modifier les résultats de l'étude. Elle est de ce fait adaptée à la situation où le chercheur est par exemple le concepteur d'un artefact dans un contexte industriel. Sa position

d'acteur et d'observateur privilégié lui permet d'analyser finement tout phénomène lié à l'artéfact en question et d'y ajuster la conception de celui-ci.

La **revue systématique de littérature** n'a pas – *a priori* – vocation à être considérée comme une méthode de recherche empirique. Néanmoins, c'est bien une *méthode* étant donné qu'elle doit être conduite de manière rigoureuse et selon un protocole bien défini (Kitchenham et Charters, 2007). De plus, elle relève bien de l'empirisme car les publications sélectionnées sont analysées en tant qu'observations empiriques nécessitant organisation et interprétation. À ce titre, l'ouvrage de Wohlin *et al.* (2012) lui consacre un chapitre spécifique. La portée d'une revue de la littérature est exploratoire lorsque le chercheur cartographie les études disponibles sur une problématique dans ce qu'on appelle une *mapping study* (Petersen *et al.*, 2008). Elle devient explicative lorsque le chercheur analyse et *synthétise* les connaissances disponibles pour en tirer une conclusion précise (Assar, 2013).

Les **leçons apprises** désignent une méthode empirique inductive peu formalisée où le chercheur synthétise l'expérience acquise durant un ou plusieurs projets. Cette synthèse est exprimée de manière narrative ; elle doit cependant s'appuyer sur un entrelacement d'observations qualitatives et quantitatives recueillies par le chercheur lui-même ou rapportées par les acteurs du projet dans différents types de livrables, ou encore extraites à partir d'entretiens avec ces acteurs.

L'**analyse Post-Mortem** (*legacy*)³ est similaire à la méthode d'observation de projets, elle se distingue cependant par l'analyse *a posteriori*, une fois que les projets sont terminés. C'est aussi une variante de la méthode des *leçons apprises*, dans la mesure où le chercheur va tenter d'apprendre des expériences passées pour synthétiser une connaissance nouvelle (Stålhane *et al.*, 2003).

La **fouille de données** (*data mining*)⁴ est une approche quantitative d'analyse pour induire des connaissances nouvelles par application des techniques d'apprentissage statistique (*machine learning*). Avec la disponibilité croissante de données sur les projets informatiques, en particulier les projets de logiciels libres, cette méthode s'est beaucoup développée dans le génie logiciel (Xie *et al.*, 2009). L'intérêt particulier de la fouille de données réside dans sa puissante capacité à mettre en évidence des motifs génériques (*pattern*). Elle peut ainsi être utilisée de manière exploratoire pour décrire et explorer les données d'un phénomène sans savoir précisément ce que l'on cherche. La fouille de données peut néanmoins avoir une portée plus explicative avec les techniques *prédictives* d'apprentissage supervisé. Ces techniques élaborent par induction à partir des données un modèle prédictif qui sera ensuite testé sur d'autres données. Enfin, la fouille peut s'appliquer sur des données particulières telles que graphes ou processus. La fouille de graphes

3. Notons ici que nous avons réuni les méthodes « legacy » de Zerkowitz *et al.* (1998) et « l'analyse post-mortem » de Stålhane *et al.* (2003).

4. La fouille de donnée est mentionnée en tant que *technique d'analyse* pour la méthode « legacy » dans Zerkowitz *et al.* (1998), et non pas en tant que méthode.

exploite la théorie des graphes pour résoudre des problèmes complexes tels que la détection de communautés (Fortunato, 2010). Dans la fouille de processus, les traces d'exécution des processus métiers au sein d'une organisation sont analysées pour extraire des modèles de processus et identifier les chemins suivis dans la réalité ou vérifier la conformité avec les modèles de processus prescrits (van der Aalst, 2010).

L'**expérience contrôlée** est la méthode empirique la plus aboutie lorsqu'il s'agit d'élaborer une explication à un phénomène. L'expérience contrôlée met en jeu un modèle que l'expérimentation va mettre à l'épreuve et tester statistiquement (cf. figure 3 et §2.2). Les expérimentations nécessitent généralement un échantillon d'utilisateurs, qui pour la validité de la méthode, doit être aléatoire. Si la composition de l'échantillon n'est que partiellement aléatoire, l'expérience contrôlée est une *quasi-expérience*. C'est le cas par exemple avec deux groupes déjà formés d'élèves dans un cours ou deux équipes de développeurs dans un projet, et dont la composition quoique aléatoire, n'est pas contrôlée par le chercheur. Le pouvoir explicatif de cette méthode réside non seulement dans la diversité des tests statistiques applicables pour analyser et interpréter les résultats (Wohlin *et al.*, 2012, chap. 10), mais aussi dans les multiples possibilités qu'offre le contrôle du chercheur sur le déroulement de l'expérimentation et les hypothèses qu'il met en jeu. Cette articulation entre l'objectif de recherche et l'organisation de l'expérience est déterminante pour la validité de l'évaluation et l'irréfutabilité de la démonstration escomptée. Il est recommandé par exemple d'utiliser un groupe de contrôle pour pouvoir comparer l'effet *avec* et *sans* l'artéfact (Hevner et Chatterjee, 2010b, p. 117).

L'**analyse comparative** est une variante de la méthode d'assertion. L'artéfact est testé sur un jeu de données prédéfini et reconnu par une communauté ou un organisme indépendant que l'on appelle un *benchmark*. C'est une méthode particulièrement adaptée à l'évaluation selon des critères quantitatifs liés à la performance (Desprez *et al.*, 2012, p. 7). Dans l'ingénierie des SI, certains domaines de recherche ont vu émerger des initiatives pour établir une évaluation d'artéfacts à base de benchmark. A titre d'exemples, on peut citer l'initiative du groupe CRIS de l'IFIP pour comparer des méthodes d'ingénierie à l'aide d'un même cas d'étude (Olle *et al.*, 1982), ou la démarche similaire de la communauté d'ingénierie des langages pour évaluer rigoureusement plusieurs méta-outils (Erdweg *et al.*, 2015).

La **simulation** est une méthode d'évaluation bien connue dans les domaines classiques d'ingénierie. Elle exploite des modèles mathématiques qui reproduisent par calcul certains aspects du comportement dynamique d'un système complexe. Transposée dans l'ingénierie des SI, cette méthode peut se combiner avec la fouille de données ou l'analyse comparative à base de benchmark pour explorer le comportement d'un artéfact selon des scénarios d'usage. C'est une méthode adaptée à l'évaluation d'artéfacts et l'exploration de problématiques liées aux processus métiers et aux processus d'ingénierie (Müller et Pfahl, 2008). Elle peut aussi avoir une portée explicative lorsque la simulation incorpore des hypothèses que le chercheur souhaite tester pour établir des corrélations ou des liens de cause à effet.

Quelle méthode choisir ?

Ce bref tour d'horizon met en évidence la diversité des méthodes disponibles pour réaliser une recherche empirique et évaluer un artéfact. Au-delà des trois méthodes les plus connues et les mieux documentées (enquête, étude de cas et expérience contrôlée), de multiples variantes et combinaisons sont possibles. On illustrera cette diversité dans la prochaine section à l'aide de plusieurs exemples.

Devant cette multiplicité des approches et lorsqu'il est face à une problématique donnée ou qu'il dispose d'une série particulière d'observations empiriques, le chercheur est confronté au problème du choix de la méthode à adopter. D'après Easterbrook *et al.* (2008), le critère premier est *la question de recherche*. Il identifie ainsi dix questions types selon l'orientation descriptive, exploratoire ou explicative de la recherche. Il insiste aussi sur la nature épistémologique de la connaissance obtenue et la validité de la réponse empirique liée au paradigme scientifique sous-jacent⁵ (*ibid.*, p. 291). Pour encore mieux guider le chercheur dans son choix, Wohlin et Aurum (2015) définissent un processus très élaboré de prise de décision. Ce processus identifie trois phases dans une étude empirique (stratégique, tactique, opérationnelle) et contient huit points de décision relatifs à diverses caractéristiques de l'étude tels que le résultat attendu (fondamental ou appliqué), la logique (déductive ou inductive), l'objectif (décrire, explorer, etc.), l'approche épistémologique, le processus (quantitatif, qualitatif), etc.

4. Illustrations de l'usage des méthodes empiriques

Pour illustrer l'utilisation des méthodes que nous venons de présenter, nous allons analyser un échantillon de neuf études empiriques. Dans la mesure du possible, le choix de ces études a été fait de manière à couvrir un grand nombre de méthodes, et à illustrer le potentiel et la complémentarité de ces méthodes dans les démarches d'investigation. Pour chaque étude (cf. tableau 2), outre un bref énoncé de la problématique traitée et l'intitulé des méthodes utilisées, nous faisons référence au cadre de Hevner des sciences de conception (cf. figure 2) et indiquons le(s) cycle(s) de recherche dans le(x)quel(s) se situe la contribution de l'étude. De plus, nous indiquons selon les cas si la portée de l'étude est d'explorer, d'expliquer ou les deux à la fois (cf. §2.4). Ces indications sont établies selon notre propre analyse et compréhension de l'étude. Enfin, nous indiquons le terrain empirique à partir duquel l'étude a été réalisée.

4.1. De l'observation à l'expérimentation

Plusieurs études illustrent cette progression fondamentale dans la recherche empirique qui va d'une observation analytique des usages et des pratiques vers une

5. Positivisme, constructivisme, pragmatisme, théorie critique.

expérimentation qui cherche à confirmer ou infirmer ces observations. Dans Davies *et al.* (2006), l'analyse textuelle des réponses aux questions ouvertes de l'enquête auprès de praticiens met en évidence – entre autres – la complexité des modèles comme l'un des obstacles à l'usage de la modélisation conceptuelle. L'étude de Mendling *et al.* (2007) s'attache à analyser plus en profondeur la complexité d'un modèle de processus et à étudier les possibles facteurs susceptibles d'influer sur la perception de cette complexité. L'expérimentation établira un lien clair entre le nombre moyen d'arcs dans un nœud d'un diagramme et complexité perçue.

Les deux études d'Engelsman et Wieringa (2012 ; 2014) font référence à un langage de modélisation d'entreprise étendu avec des concepts issus de l'ingénierie des besoins, notamment le concept de but. La première étude explore la facilité d'utilisation et l'utilité du langage au travers de projets conduits par un des auteurs, sous forme de recherche-action au sein de deux organisations. L'étude conclut que le concept de but est perçu comme utile pour assurer la traçabilité entre les objectifs métiers et l'architecture du SI. Néanmoins, il apparaît que certains concepts sophistiqués de l'ingénierie des besoins ne sont pas utilisables en entreprise car difficiles à comprendre. La seconde étude cherche à confirmer ce résultat et à évaluer la compréhensibilité de ces concepts. Les résultats démontrent *empiriquement* qu'effectivement, malgré l'expertise affirmée de participants, plusieurs concepts sont utilisés incorrectement car mal compris.

Enfin, l'ingénierie dirigée par les modèles (IDM) soulève beaucoup d'espoirs pour maîtriser la complexité des logiciels et systématiser leur développement. Néanmoins, peu d'études empiriques sur le terrain le confirment. Dans trois études de cas, l'analyse de Hutchinson *et al.* (2011) combine l'observation minutieuse des processus de développement avec l'interprétation qualitative du discours des acteurs. Elle met en exergue la dimension organisationnelle dans le succès de l'IDM, avec notamment la nécessité de gérer le changement (de l'organisation, des méthodes de développement, des outils, etc.) et d'adopter une démarche flexible qui favorise le dialogue et le développement des compétences. Cependant, l'apport concret de l'IDM reste inconnu. L'étude de Panach *et al.* (2015) examine cette question à travers une expérimentation contrôlée, avec deux groupes d'étudiants développant à tour de rôle deux applications similaires, l'une avec un langage de programmation classique et l'autre avec un outil de modélisation et de génération de code. Les résultats ne permettent pas de conclure à un avantage significatif pour l'IDM.

Tableau 2. Analyse d'un ensemble d'études empiriques

	Thème et problématique	Méthode(s)	Cycle de Hevner	Portée	Terrain
Modélisation	(Davies <i>et al.</i> , 2006) – quelle utilisation concrète de la modélisation conceptuelle ?	– Enquête – Entretiens semi-structurés	Cycles 2 et 3	Explorer et expliquer	– 312 praticiens – 980 commentaires
	(Mendling <i>et al.</i> , 2007) – les modèles BPMN sont-ils aisément compréhensibles ?	– Quasi-expérience contrôlée – Enquête	Cycle 3	Expliquer	– 73 étudiants dans 3 universités – 847 observations
Ing. des besoins	(Engelsman et Wieringa, 2012) – est-ce que la modélisation de but améliore la traçabilité entre architecture et objectifs métiers ?	– Étude de cas – Recherche action – Leçons apprises	Cycles 1 et 2	Explorer	– une organisation gouvernementale – une entreprise industrielle
	(Engelsman et Wieringa, 2014) – les modèles d'entreprise enrichis avec le concept de but (entre autres) sont-ils compréhensibles ?	– Quasi-expérience contrôlée	Cycles 2 et 3	Explorer et expliquer	– deux groupes de 7 et 12 praticiens
IDM	(Hutchinson <i>et al.</i> , 2011) – quelle perception de l'IDM dans l'industrie ?	– Étude de cas – Ethnographie	Cycle 2	Explorer	– trois entreprises industrielles – 20 praticiens
	(Panach <i>et al.</i> , 2015) – est-ce que l'IDM contribue à réduire l'effort requis ou améliorer la productivité ou la satisfaction ?	– Expérience contrôlée – Enquête	Cycle 3	Expliquer	– 26 étudiants – un outil d'IDM – deux exercices similaires
Génie logiciel	(Harter <i>et al.</i> , 2012) – est-ce que l'amélioration du processus de développement selon CMMI réduit la sévérité des erreurs logicielles ?	– Analyse <i>post-mortem</i> – Étude longitudinale de terrain	Cycles 2 et 3	Explorer et expliquer	– un projet de développement logiciel sur ~20 ans – 7,545 erreurs logicielles
	(Assar <i>et al.</i> , 2015) – est-ce que la technique de classification non supervisée permet de prédire le temps de résolution d'une erreur logicielle ?	– Fouille de texte – Simulation > <i>Réplication de (Raja, 2013)</i>	Cycles 1 et 3	Explorer et expliquer	– 3 jeux de données – 15,632 commentaires
	(Foucalt <i>et al.</i> , 2015) – est-ce que la paternité du code influe sur sa qualité ?	– Analyse <i>post-mortem</i> > <i>Réplication de (Bird et al., 2011)</i>	Cycle 3	Explorer et expliquer	– 7 jeux de données – 384 modules – 3 catégories d'observations

4.2. L'abduction à l'œuvre

Avec l'abduction, le chercheur émet des hypothèses explicatives au phénomène observé et vérifie les conséquences de ces hypothèses à l'aide d'observations et d'expériences. L'étude de Harter *et al.* (2012) est intéressante à ce titre : les auteurs font l'hypothèse que l'élévation du niveau de maturité de l'organisation en termes de développements informatiques (mesurée par le modèle CMMI, *Capability Maturity Model Integration*), contribue à la réduction des erreurs dans les logiciels produits. Cette étude purement observationnelle a néanmoins une portée explicative grâce au grand volume de données récoltées et une longue période d'observation, période pendant laquelle le niveau de maturité de l'organisation a substantiellement progressé. Le résultat obtenu est positif, il est néanmoins circonscrit aux erreurs les plus sévères : à un niveau élevé de maturité, il y a toujours autant d'erreurs, cependant, les erreurs sévères sont moins nombreuses. Ce constat est validé statistiquement, et les auteurs concluent qu'un niveau élevé de maturité aboutit à des processus de développement logiciel de meilleure qualité.

4.3. De l'induction à la déduction

La fouille de données permet d'induire des connaissances nouvelles à l'aide des techniques d'apprentissage machine (cf. §3). Néanmoins, il est prudent de considérer que l'induction, qu'elle résulte d'observations humaines ou de calculs informatiques, relève du domaine exploratoire et que les connaissances produites par induction doivent être validées par de nouveaux tests. Dans Assar *et al.* (2015), les auteurs examinent une règle empirique obtenue par fouille de texte concernant l'estimation du temps de résolution d'une erreur logicielle. Pour évaluer la validité prédictive de cette règle, ils simulent son application sur plusieurs jeux de données et selon des scénarios multiples. La comparaison entre les prédictions de cette règle et celles d'une technique plus basique laissent penser que la règle n'est pas utile.

L'étude de Foucault *et al.* (2015) adopte la même démarche de déduction à partir d'une règle observée empiriquement. La règle exprime une corrélation entre une paternité forte pour un code et sa qualité (mesurée par le taux d'erreurs). La paternité fait référence à plusieurs métriques relatives au nombre de développeurs ayant contribué à la réalisation et la maintenance d'un code. Est-ce que cette règle améliore les modèles de prédiction existants ? Les résultats démontrent que la règle est valide (la corrélation est démontrée statistiquement), néanmoins, sa capacité de prédiction est faible lorsqu'elle est comparée aux modèles existants.

4.4. Réplicabilité et accumulation des connaissances

Les deux études discutées dans la section précédente sont des répliques non exactes de travaux publiés antérieurement. Plusieurs modifications ont été apportées

pour élargir le périmètre de validité des études initiales et mettre en évidence d'éventuels facteurs confondants. Dans les deux cas, les résultats sont confirmés et contribuent ainsi à renforcer la validité de ces études. Néanmoins, ces résultats n'étant pas exploitables, ces répliques illustrent la difficulté à obtenir des connaissances empiriques utiles et significatives pour l'avancée de la recherche.

5. Conclusion

Dans les sciences de la nature, l'empirisme en tant qu'observation critique et expérimentation confondante, fut le point de départ de plusieurs grandes découvertes scientifiques. En ingénierie, la logique des découvertes est un peu différente. C'est la maîtrise des lois qui gouvernent le fonctionnement interne des artefacts qui permet, par *application innovante* de ces lois, de créer une multitude d'artefacts. Les seules limites sont l'imagination créative de l'Homme et l'applicabilité concrète de ces lois. Cependant, quelle utilité, quelle applicabilité, bref, quel *avenir* pour ces artefacts ? C'est là où l'empirisme trouve sa place dans les sciences de l'ingénieur.

L'éventail des méthodes empiriques est très large, et la pratique consiste souvent à combiner plusieurs méthodes ou à adopter une variante particulière. Nous avons passé en revue l'ensemble de ces méthodes telles qu'elles apparaissent dans les principales publications en ingénierie informatique. La terminologie n'est pas toujours homogène, et il est parfois difficile de faire la distinction entre la nature de la donnée, le protocole de récolte, la technique d'analyse ou du paradigme épistémologique sous-entendu. C'est le cas notamment avec les termes *qualitatif* et *quantitatif* qui, par abus de langage, peuvent désigner chacun de ces aspects.

L'échantillon de travaux que nous avons choisi d'analyser a permis d'illustrer la démarche empirique d'investigation que mobilise le chercheur pour évaluer un artefact technologique. Nous avons pris soin de mettre en évidence l'importance des deux perspectives : explorer un phénomène ou l'expliquer. Si l'explication aboutit à une connaissance conceptuelle ou théorique nouvelle, l'observation permet néanmoins de suggérer des pistes pour de nouvelles recherches et pour découvrir de nouveaux artefacts grâce notamment à l'observation inductive. A terme, le passage de l'exploration à l'explication dépendra de la richesse des données recueillies et de la rigueur dans la conduite de la démarche d'investigation. Cette rigueur est un vrai défi pour le chercheur car les pratiques collectives et les normes de publication de la recherche empirique ne sont pas encore très répandues (Kitchenham *et al.*, 2002).

Quelles orientations pour la recherche empirique à venir ? D'après Basili (2013), un enjeu majeur est l'accumulation des connaissances obtenues empiriquement pour alimenter et élargir le corpus existant. L'auteur évoque la réplique comme moyen premier pour atteindre cet objectif et insiste sur le besoin de répliques qui étendent les connaissances et ne se limitent pas à les confirmer. Une autre voie dans la même direction est la synthèse des connaissances à travers notamment les revues de littérature et l'agrégation statistique des résultats empiriques (la méta-analyse).

Bibliographie

- van der Aalst W. M. P. (2010). Process Discovery: Capturing the Invisible. *IEEE Computational Intelligence Magazine*, vol. 5, n° 1, p. 28-41.
- Assar S. (2013). Pour une revue rigoureuse et systématique de la littérature en MIS: étude comparative et multidisciplinaire. *Actes du 1^{er} atelier sur les Méthodes de Recherche en Systèmes d'Information (RMiSI3)*, Nantes, France.
- Assar S., Borg M., Pfahl D. (2015). Using text clustering to predict defect resolution time: a conceptual replication and an evaluation of prediction accuracy. *Empirical Software Engineering*, p. 1-39, <http://link.springer.com/article/10.1007/s10664-015-9391-7>
- Basili V. R. (2013). A Personal Perspective on the Evolution of Empirical Software Engineering. Dans : Münch J, Schmid K (éd.). *Perspectives on the Future of Software Engineering*, Springer, Berlin/Heidelberg. p. 255-273.
- Basili V. R., Caldiera G., Rombach H. D. (1994). *Goal question metric paradigm. Encyclopedia of Software Engineering*.
- Basili V. R., Selby R. W., Hutchens D. H. (1986). Experimentation in software engineering. *IEEE Transactions on Software Engineering*, vol. SE-12, n° 7, p. 733-743.
- Bird C., Nagappan N., Murphy B., Gall H., Devanbu P. (2011). Don't touch my code!: examining the effects of ownership on software quality. *Proc. 19th ACM SIGSOFT Symposium and the 13th European Conf. on Foundations of Software Eng.*, p. 4-14.
- Brooks A., Roper M., Wood M., Daly J., Miller J. (2008). Replication's Role in Software Engineering. *Guide to Advanced Empirical Software Engineering*, Shull F, Singer J, Sjøberg DIK (éd.). Springer, London, UK. p. 365-379.
- Bunge M. (1967). *Scientific Research 1: The Search for System*. New York: Springer, [s.l.].
- David A. (1998). *Logique, épistémologie et méthodologie en sciences de gestion*. Cahier de recherche, n°. DMSP-C - 99-275, Université Paris Dauphine, France.
- Davies I., Green P., Rosemann M., Indulska M., Gallo S. (2006). How do practitioners use conceptual modeling in practice? *Data & Knowledge Eng.*, vol. 58, n° 3, p. 358-380.
- Davis F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, vol. 13, n° 3, p. 319-340.
- Denning P. J. (2013). The science in computer science. *Communications of the ACM*, vol. 56, n°5, p. 35-38.
- Desprez F., Fox G., Jeannot E., Keahey K. et al. (2012). *Supporting Experimental Computer Science*. n°. RR-8035, INRIA.
- Douven I. (2011). *Abduction*. Stanford Encyclopedia of Philosophy, <http://plato.stanford.edu/>.
- Easterbrook S., Singer J., Storey M.-A., Damian D. (2008). Selecting Empirical Methods for Software Engineering Research. *Guide to Advanced Empirical Software Engineering*, Shull F, Singer J, Sjøberg DIK (Ed.). Springer, London, UK, p. 285-311.

- Engelsman W., Wieringa R. (2012). Goal-Oriented Requirements Engineering and Enterprise Architecture: Two Case Studies and Some Lessons Learned. *Requirements Eng.: Foundation for Software Quality*, Regnell B., Damian D. (Ed.), Springer, p. 306-320.
- Engelsman W., Wieringa R. (2014). Understandability of Goal-Oriented Requirements Engineering Concepts for Enterprise Architects. *CAiSE'14*, Jarke M., Mylopoulos J. *et al.* (Ed.). Springer International Publishing, p. 105-119.
- Erdweg S., Van der Storm T., Völter M., Tratt L., Bosman R. *et al.* (2015). Evaluating and comparing language workbenches: Existing results and benchmarks for the future. *Computer Languages, Systems & Structures*, vol. 44, Part A, p. 24-47.
- Fortunato S. (2010). Community detection in graphs. *Physics Reports*, vol. 486, n° 3-5, p. 75-174.
- Foucault M., Teyton C., Lo D., Blanc X., Falleri J.-R. (2015). On the usefulness of ownership metrics in open-source software projects. *Inf. and Soft. Technology*, vol. 64, p. 102-112.
- Gómez O. S., Juristo N., Vegas S. (2014). Understanding replication of experiments in software engineering: A classification. *Inf. and Softw. Techn.*, vol. 56, n° 8, p. 1033-1048.
- Harter D. E., Kemerer C. F., Slaughter S. (2012). Does Software Process Improvement Reduce the Severity of Defects? A Longitudinal Field Study. *IEEE Transactions on Software Engineering*, vol. 38, n° 4, p. 810-827.
- Hatton L. (1998). Does OO sync with how we think? *IEEE Software*, vol. 15, n° 3, p. 46-54.
- Hevner A. R., March S. T., Park J., Ram S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, vol. 28, n° 1, p. 75-105.
- Hevner A., Chatterjee S. (2010a). Design Science Research in Information Systems. *Design Research in Information Systems*, Hevner A, Chatterjee S (Ed.). Springer, p. 9-22.
- Hevner A., Chatterjee S. (2010b). Evaluation. *Design Research in Information Systems*, Hevner A, Chatterjee S (Ed.), Springer, p. 109-119.
- Hutchinson J., Rouncefield M., Whittle J. (2011). Model-driven engineering practices in industry. *Proceedings 33rd Int. Conf. on Software Eng. (ICSE'11)*, p. 633-642.
- Juristo N., Gómez O. S. (2012). Replication of Software Engineering Experiments. *Empirical Software Engineering and Verification*, Meyer B, Nordio M. (Ed.). Springer, Berlin/Heidelberg, p. 60-88.
- Juristo N., Vegas S. (2009). Using differences among replications of software engineering experiments to gain knowledge. *Proceedings 3rd Int. Symposium on Empirical Software Eng. and Measurement (ESEM'09)*, IEEE Computer Society, p. 356-366.
- Juristo N., Vegas S. (2010). Replication, Reproduction and Re-analysis: Three ways for verifying experimental findings. *Proceedings 1st Int. Workshop on Replication in Empirical Software Eng. Research (RESER 2010)*, Cape Town, South Africa.
- Kitchenham B. A. (2008). The role of replications in empirical software engineering. A word of warning. *Empirical Software Engineering*, vol. 13, n° 2, p. 219-221.
- Kitchenham B. A., Charters S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering - Version 2.3*. EBSE Technical Report, Keele Univ.

- Kitchenham B. A., Dybå T., Jørgensen M. (2004). Evidence-based software engineering. *Proceedings 26th Int. Conf. on Software Engineering (ICSE)*, IEEE Computer Society, Washington DC, USA. p. 273-281.
- Kitchenham B. A., Pfleeger S. L., Pickard L. M., Jones P. W., Hoaglin D. C., El Emam K., Rosenberg J. (2002). Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, vol. 28, n° 8, p. 721-734.
- Kitzinger J. (1995). Qualitative research: introducing focus groups. *British Medical Journal*, vol. 311, n° 7000, p. 299-302.
- Kontio J., Bragge J., Lehtola L. (2008). The Focus Group Method as an Empirical Tool in Software Engineering. *Guide to Advanced Empirical Software Engineering*, Shull F., Singer J., Sjøberg DIK (Ed.). Springer London, p. 93-116.
- Larousse. (2015). *Dictionnaire de Français*. Édition en ligne.
- Lecourt D. (2012). *La philosophie des sciences*. 5^e édition. Presses Universitaires de France, coll. *Que sais-je ?*, Paris.
- McLeod L., MacDonell S., Doolin B. (2011). Qualitative research on software development: a longitudinal case study methodology. *Empirical Softw. Eng.*, vol. 16, n° 4, p. 430-459.
- Mendling J., Reijers H., Cardoso J. (2007). What Makes Process Models Understandable?. *Business Process Management*, Alonso G. et al. (éd.). Springer, p. 48-63.
- Müller M., Pfahl D. (2008). Simulation Methods. *Guide to Advanced Empirical Software Engineering*, Shull F., Singer J., Sjøberg DIK (Ed.), Springer London, p. 117-152.
- Olle T. W., Stuart A. A. V., Sol H. G. (1982). *Information Systems Design Methodologies. A Comparative Review*. Elsevier Science Inc., New York, NY, USA.
- Panach J. I., España S., Dieste Ó., Pastor Ó., Juristo N. (2015). In search of evidence for model-driven development claims: An experiment on quality, effort, productivity and satisfaction. *Information and Software Technology*, vol. 62, p. 164-186.
- Petersen K., Feldt R., Mujtaba S., Mattsson M. (2008). Systematic mapping studies in software engineering. *12th Int. Conf. on Evaluation and Assessment in Software Eng. (EASE'08)*, p. 71-80.
- Pfleeger S. L., Kitchenham B. A. (2001). Principles of Survey Research: Part 1: Turning Lemons into Lemonade. *SIGSOFT Softw. Eng. Notes*, vol. 26, n° 6, p. 16-18.
- Prat N., Comyn-Wattiau I., Akoka J. (2014). Artifact Evaluation in Information Systems Design-Science Research. A Holistic View. *PACIS 2014*, AIS Electronic Library.
- Raja U. (2013). All complaints are not created equal: text analysis of open source software defect reports. *Empirical Software Engineering*, vol. 18, n° 1, p. 117-138.
- Rosenthal R. (1991). Replication in Behavioral Research. Neuliep J.W. (Ed.). *Replication Research in the Social Sciences*, SAGE Publications Inc, p. 1-30.
- Runeson P., Höst M., Rainer A., Regnell B. (2012). *Case Study Research in Software Engineering*. John Wiley & Sons, Inc.

- Santos P., Travassos G. H. (2009). Action research use in software engineering: An initial survey. *3rd Int. Symposium on Empirical Software Eng. and Measurement (ESEM'09)*, p. 414-417.
- Sein M., Henfridsson O., Purao S., Rossi M., Lindgren R. (2011). Action Design Research. *MIS Quarterly*, vol. 35, n° 1, p. 37-56.
- Shull F., Carver J., Vegas S., Juristo N. (2008). The role of replications in Empirical Software Engineering. *Empirical Software Engineering*, vol. 13, n° 2, p. 211-218.
- Shull F., Singer J., Sjøberg D. I. K. (2008). *Guide to Advanced Empirical Software Engineering*. Springer, London, UK.
- Simon H. A. (1969). *The Science of the Artificial*. 1st edition. MIT Press, Cambridge.
- Sofaer S. (1999). Qualitative methods: what are they and why use them? *Health services research*, vol. 34, n° 5 Pt 2, p. 1101-1118.
- Stålhane T., Dingsøyr T., Hanssen G. K., Moe N. B. (2003). Post Mortem – An Assessment of Two Approaches. *Empirical Methods and Studies in Software Engineering*, Conradi R., Wang A.I. (Ed.). Springer, Berlin/Heidelberg. p. 129-141.
- Tichy W. F. (1998). Should computer scientists experiment more? *Computer*, vol. 31, n° 5, p. 32-40.
- Tichy W. F., Lukowicz P., Prechelt L., Heinz E. A. (1995). Experimental evaluation in computer science: A quantitative study. *Journal of Sys. and Softw.*, vol. 28, n° 1, p. 9-18.
- Weick K. E. (1995). What Theory is Not, Theorizing Is. *Administrative Science Quarterly*, vol. 40, n° 3, p. 385-390.
- Wohlin C., Aurum A. (2015). Towards a decision-making structure for selecting a research design in empirical software engineering. *Empir Softw Eng*, vol. 20, n° 6, p. 1427-1455.
- Wohlin C., Höst M., Henningsson K. (2003). Empirical research methods in software engineering. *Empirical Methods and Studies in Software Engineering: Experiences from ESERNET*, Conradi R, Wang AI (Ed.), Springer, Berlin/Heidelberg. p. 7-23.
- Wohlin C., Runeson P., Höst M., Ohlsson M. C., Regnell B., Wesslén A. (2012). *Experimentation in Software Engineering*. Springer, Berlin/Heidelberg.
- Xie T., Thummalapenta S., Lo D., Liu C. (2009). Data Mining for Software Engineering. *Computer*, vol. 42, n° 8, p. 55-62.
- Zelkowitz M. V., Wallace D. (1997). Experimental validation in software engineering. *Information and Software Technology*, vol. 39, n° 11, p. 735-743.
- Zelkowitz M. V., Wallace D. R. (1998). Experimental models for validating technology. *Computer*, vol. 31, n° 5, p. 23-31.

