Testing and Music: Parallels in Practice, Skills, and Learning

For years, people have appealed to engineering as the dominant metaphor for how software testing should be done. Yet there are many other metaphors that could provide value to learning about patterns and principles in software testing. As professional testers and non-professional musicians, we observe that music affords such a metaphor. What are the parallels between music and testing—especially exploratory approaches to testing? What can we learn from the similarities?

In this paper, we explore traditions and contexts; structures of performance in music and testing; ideas associated with tension and resolution; the role of scripting and other artifacts in design, performance, and learning; and skills development.

Traditions and Contexts

Both testing and music are performed in a wide variety of contexts, with different audiences, different practitioners, and different values for different people. Both fields involve, to a large degree, socially constructed activities, and are suffused with traditions.

Living traditions depend on tension between three points of view: a classical aspect, which preserves the foundation of the practice, but which views change and diversity as a threat to the purity of the art; a state of the practice, which advances slowly while absorbing some forms of change and resisting others; and an avant-garde, which stretches the limits and boundaries of the state of the art by bringing in influences from outside and synthesizing new forms, but which may not feel beholden to the classical foundations. The classicists and the avant-garde tend to argue with one another, while the middle ground simply proceeds without paying too much attention to the other extremes.

In music, cultures and subcultures abound, cross-pollinating one another. For example, Irish traditional dance music, a subculture of Celtic traditions that also include Scottish, Breton, Welsh, Cape Breton, has at least four dominant sub-styles (Clare, Sligo, Donegal, and Cork/Sliabh Luachra). The blues has Chicago, Memphis and Mississippi Delta traditions, as well as boogie woogie, country blues and blues rock. One example of a veritable hotbed of colliding musical styles is Zydeco, a style of music created in Louisiana. Zydeco began as a fusion of Creole, Cajun (from "Acadian"—itself a blend of French and Celtic styles from Canada's east cost) and traditional American music. It has further evolved to include influences from blues, jazz, gospel, and other popular North American music as well as Caribbean influences such as salsa, rumba and calypso, among others. It's not uncommon to hear hip hop styling or steel drums behind the driving accordion that is a trademark of the genre.

It's important for both testers and musicians to perform in a manner appropriate to the context. Electronic dance music tends not to go over well in a church that favours Gregorian plainchant for its liturgical music. On the other hand, in a supportive context, musical styles can blend, leading to fusion and innovation that can advance new traditions and recall old ones. The startlingly successful Enigma recordings of the early 1990s may have contributed to the success of the fusion of plainsong and chant with dance music during that decade.

Testing has been (somewhat controversially) categorized into schools [Pettichord2003]. We assert that these schools can be seen as analogous to musical traditions. Schools of testing can be strengthened by adding techniques and models associated with other schools. In software testing, opponents to the schools categorization often claim that all testing styles are the same, and that division is unnecessary. In music, diversity is embraced and encouraged, and has led to discovery and growth. In testing, we can also profit from identifying different ideas, styles and interpretations. The recognition of specialization and focuses from each of these areas will lead to more discovery, communities of practice, and growth for the whole software testing community. To deny the differences of ideas in testing is akin to saying that music is only "music" and any attempt to identify differing styles, traditions and genres is divisive and problematic.

Structures

A musical piece has structure whether it is rehearsed or improvised, played from a score or played extemporaneously. Several elements of the piece's structure are determined by time. The *tempo* of the piece refers to its speed or pace. *Rhythm* refers to the duration of a series of notes and the ways in which they are grouped together. A *bar*, or *measure*, is a means of dividing the rhythm into sequences of beats; in most popular Western music, bars in "straight time" tend to have four beats each (bars in "waltz time" have three); these associations and divisions give the piece its *meter*. *Loudness* and *accents* give *dynamics* to a piece. Other structural elements are determined by *tones*; the *pitch* (or frequency) of the notes in the piece; the *key* of the piece (the tonic or harmonic centre of the piece, that forms the base of a scale) and the *mode*, which specifies the intervals between each note in the scale. *Melody* (the tune, sometimes known as the *air* in traditional music) is the fundamental theme of the piece. *Chords*—combinations of three or more notes played at the same time—and *contour*—changes or progressions from one note or chord to the next—also give structure to a piece. We only scratch the surface here.[1]
Blues music has a very common pattern of chords, expressed as the "twelve bar blues". (The Lieber and Stoller song "Kansas City" is an example of this pattern; so is their song "Hound Dog", best known as performed by Elvis Presley.) In a jam session, someone need only say "twelve bar blues, in (the key of) A", and everyone will have a common framework in which to play. Experienced musicians will need no other direction than that to begin playing music together. Improvisations and variations on the structure give richness and interest to the music.

Irish traditional music also has very common structures. Reels have four beats to the bar; jigs have six. (An easy way to keep them straight: the jig has the rhythm "jiggedy, jigeddy", and the reel doesn't; it has a rhythm that can be approximated by saying "wish I had a motorcycle".) Tunes are usually played as a pattern of "aabb"—eight bars with a given melodic line (a), a second eight bars that repeat that line or provide a slight variation (also (a); these two lines together are comprise the "A section"); and then eight bars with a different melodic line (b), and a second eight bars that recall that line and resolve toward the tonic or root of the chord (the latter two lines are the "B section"). In a "set" the players tend to play several tunes (typically three), playing each tune three times through and the switching to another tune, generally in the same rhythm, although the key may change (or "modulate") between tunes. Irish music has very

[1] An excellent description of these elements of musical structure can be found in Levitin, *This Is Your Brain On Music*. [Levitin2006]

strong and intricate melodies. Tunes tend to be in the keys of D, G, or A, since these are keys that are relatively easy to play on the fiddle, the flute, and the tin whistle, instruments that dominate the genre.

Remarkably, many capable musicians have a limited knowledge of musical theory. Nonetheless, these structures are sufficiently powerful that non-theorists can perform music just fine. That is, they appear to have an intuitive grasp of the structures. A regular participant in Irish traditional music sessions will know hundreds of tunes; a blues player will adapt easily to the many variations on the basic themes of the genre. Yet these people may have had little or no formal musical training, and may not be able to read music. So how do they remember pieces? According to Levitin, they rely on a structure for their memory, and the details fit into that structure. [Levitin2006, p. 213]

Testing also has structures, both in design and performance. We would argue that exploratory testing is structured in ways that are analogous to improvisational music. As a relatively new tradition, many of the structures haven't been named or even noticed, particularly for exploratory approaches. Practitioners—even experts—sometimes have a difficult time articulating what they do. Some experts can often play well only inside their own contexts. The cognitive patterns are mysterious and we're still learning how to understand them and the ways in which they interact with other contextual elements of testing.

Some of the musical genres and styles we have introduced can also be described as *patterns* and *practices* with related *techniques.* Musicians will often learn the patterns of what sounds right in a particular genre, without knowing the theory behind what they are performing. In testing, we have observed different kinds of patterns, practices and techniques as well, and many testers have a limited knowledge of testing or computer science theory. Both of us work as testing trainers, and have noticed that many testers will immediately try similar kinds of patterns or techniques when given a particular program to test as an exercise. For example, when posed with an application that has input fields, most testers will try overflow attacks, or will try to enter in values of the wrong type. Similarly, musicians adapt when playing music, depending on the genre. Again, we scratch the surface here – there are many patterns that are often used in testing and in music.

James Bach suggests that the structure of exploratory testing comes from many sources: test design heuristics; chartering; time boxing; perceived product risks; the nature of specific tests; the structure of the product being tested; the process of learning the product; development activities; constraints and resources afforded by the project; the skills, talents, and interests of the tester; the overall mission of testing; and the testing story. [BachRST] James and Jon Bach have proposed lists of exploratory skills, tactics and dynamics that refer to patterns of performance [JamesBach2005]. Bach, Mike Kelly, Jonathan Kohl, Scott Barber, Ben Simo, and other testers have suggested mnemonics to remember guidewords heuristics. (Kohl introduces the idea of comparing music and testing mnemonics in [Kohl2007])

In the most extreme form of improvisation, avant garde musicians abandon not only score but also structure. The result is occasionally interesting, but isn't usually popular to listeners other than devotees. It is often performed as an experimental exercise in the attempt to discover

something new. Avant-garde music tends to help create new genres, and provide a space to help new ideas foster and grow. Without experimentation, discovery of new forms or new fusions of old forms can be suppressed. Testing is fundamentally experimental and investigative, and would profit greatly from avant-garde ideas both on testing projects and in the software testing community as a whole.

Since software testing is not generally performed for entertainment, or practice in the same way that performing music is, there are other areas where the analogy breaks down. Since testing is about discovering and reporting important information, combinations of practices and techniques that would be unlistenable in a musical setting are areas of discovery in testing. Since we are not usually performing for an audience who is expecting to see or hear something pleasing, we could have more room for experimentation in testing.

In music, composition and performances are often critiqued according to different standards and aesthetics. In testing, we often talk about test coverage, but we don't often evaluate how effective our testing is. There are potential lessons we could draw from musical critiques, and take form, structure, variation and diversity of approach into account as we evaluate our software testing efforts.

Tension and Resolution

In composition and performance, music often exhibits aspects of tension and resolution. A typical piece of music is written in a given key, and typically the piece ends on the tonic, the first note or root of a scale in that key, accompanied by the fundamental chord that shares the name of that key. Patterns of notes and chords in that scale create suspense that is resolved by a return to the root.

In a testing session, tension and resolution revolve around testing ideas, rather than musical notes. Tension and suspense are generated by a test idea, a question about the system under test. Resolution comes with an answer to that question, produced by operating and observing the product. We see another parallel between music and testing. Too much tension raises discomfort; too much resolution becomes boring, tedious repetition. Testers and musicians alike need to find a balance between tension and resolution, and to find this balance, they need a mix of knowledge, skill and creativity. [Kohl2007]

Feelings of tension and resolution in music are also felt and observed by the audience as they listen to a live or recorded performance. Music practitioners also learn from watching others at work. Performing music is related to the practice and skill development of a musician, and the listening enjoyment of the audience. In music, most of the information guides and is locked up in the performance. In music, impressions are generally about the qualities of the performance itself.

The elements and focus of the performance is one area where our analogy breaks down to some degree. In both testing and music, the audience derives an impression from the performance, and much of this impression is sustained after the fact. But software testers generally do not perform their work in concert halls, nor in front of audiences. Their work is conducted in relative isolation, with a different goal in mind: to gather as much important information for stakeholders

as they can. Testing is not usually done for the benefit of an audience watching the tester doing his work. Instead, the value for the audience is in the information *derived* from the performance, rather than the performance itself.

There are some exceptions on the performance issue in software testing. Some exploratory testing teachers such as James Bach, Jon Bach, Michael Bolton, Jonathan Kohl and others do live testing demonstrations. With the rise in popularity of video on the web, many are recording test sessions for the benefit and enjoyment of others—typically other testers, or people who wish to learn about testing. The difference in these performances is that they are usually done for teaching purposes, not for the viewing or listening enjoyment of a broad audience. Like musicians, testers can learn from watching others perform. Furthermore, differences in styles and genres become much more apparent when demonstrated. We see this as an opportunity for testing education.

Scripting

In both music and testing, there is a dynamic between scripted processes (in which the ideas come from some person or agency at some point in the past) and improvisational or more exploratory processes (in which ideas are created and discovered on the fly, during performance).

For a given activity, we define scripted and exploratory approaches to be at opposite ends of a continuum. In a scripted approach, the process of design and execution of the activity are separated in time, and typically in the person performing them. Some person composes, designs or synthesizes ideas in advance of the activity, and commits them to some medium—typically in a textual or written form. The person performing the activity interprets the text and is guided by those ideas. We define the degree to which an activity is scripted as the extent to which the idea and the precise steps to exercise it are specified in advance; the extent to which those ideas guide the person performing the steps; and the degree to which learning associated with design is separated from learning associated with the activity. An exploratory approach is one in which design and execution happen simultaneously, not separated either by time or by person. Instead, composition and performance happen in a way that responds to context; to the skills of the performer; to what just happened; and to a consensus, often unspoken, on what should happen next. Learning about design and learning about the activity are not separated; they too happen simultaneously.

A purely scripted approach in music is a very strict interpretation of the piece as composed, typically by reading a score. At the opposite pole from playing a piece by reading a score is playing a piece “by ear”. Music played by ear is played without sheet music or with minimal guidance from it. Instead, the musician learns the piece and its structure by listening to others play it. Playing by ear is sometimes but not always associated with improvisation, in which musicians compose and perform their ideas simultaneously. The players make choices about what to play based on the structure of the piece; skills in listening to and observing other performers; technical and physical skills; the emotions and mood of the players. Successful improvisation requires skill, and top performers study to develop a large breadth and depth of musical theory and technical proficiency on their instruments in order to successfully and

creatively improvise. A purely exploratory approach in music performance is free-form improvisation. There are many variations in between. Few musicians can achieve a purely scripted interpretation. Conversely, very few (if any) musicians have the skills and ability to only play music that is influenced by the last note that was played in free-form improvisation.

Western classical music is highly scripted in the form of a *score*. A score, or sheet music, uses a highly specific notational system that allows performers to reproduce the basis of the piece with their voices or instruments. A score typically specifies the melody—the tonal and rhythmic patterns of the notes to be played. The score may also identify harmony—other notes or chords to be played at the same time as the melody, possibly identifying different notes or countermelodies for various instruments; the tempo—the speed at which the piece is played; volume; accents; and even bow strokes. Despite this rich, detailed, well-disseminated, and shared "language" for written music, it is difficult to perform music exactly the way the composer intended. Performance on non-electronic instruments will always include variations in intonation, timbre, volume, dynamics, and embellishments. These variations might be subtly nuanced, or performed with a flourish; they might remain quite faithful to the original or common or they might be dramatic reinterpretations.

When performing a well-known piece of music, there are scripted and unscripted dynamics at play. Even when played from a score and when under the direction and supervision of the author of the music, subtle variations creep in. Reproducing the composer's ideas is particularly hard with musical pieces that have been around for centuries, because we don't have the composer around anymore to consult. Bodies of traditional interpretation tend to arise around pieces as they age. In popular or traditional music, people frequently play without any sheet music at all. This might suggest that there is a great deal of freedom in unscripted music, but this isn't always the case. In most styles of music—such as Irish traditional music—players adhere strongly to melody, even in the absence of a score. In Indian classical music, music is not recorded in scores; instead, the scripts are passed down through an oral tradition. Music students are taught by a teacher or guru in this manner: The teacher plays or sings a part, and has the student repeat it. While this teaching style differs from Western classical music where the music is written down, both are using scripted approaches: the ideas come from some time in the past, and from another person.

There are several factors that influence the decision to be faithful to the script. One factor in playing a piece from a score is the level of detail in the sheet music itself. A Western classical orchestral piece tends to be very highly scripted. Because many instruments and players must be coordinated to achieve a precisely desired effect, the individual lines of music may be laid out very specifically. Nonetheless, the performance is still strongly influenced by the individual musicians' playing styles and the interpretation of the orchestra's conductor. By contrast, in general, scoring for popular music—if used at all—tends to be less detailed. A typically arrangement of a song provides detailed music for piano, the melody line for a singer, and chords for guitar; other scores contain only the melody line and the chords.. The score itself affords the opportunity for a cognitively engaged player to bring some level of variation and interpretation to the performance. In fact, such scores mandate interpretation because they are sparsely detailed. At the other extreme, software can be programmed to play music such that it is very precise in repeating what is input from a score, but it tends to be boring and tedious, rarely as

interesting and as pleasant to listen to as real performers are. In popular music, audiences and performers alike tend to allow a lot of room for improvisation and spontaneous discovery.

Scripted and exploratory approaches to testing are similarly on opposite ends of a continuum. In a scripted approach, the processes of test design and test execution are separated by time, and typically by person performing them. A test designer develops test ideas, and records them in advance; the person performing the test is guided by these ideas. The degree to which a test is scripted is the extent to which the test idea and the steps to exercise it are specified in advance. An exploratory approach is one in which design and execution are not separated, either by time or by person. Instead, the tester performs each test in a way that can incorporate all of his or her knowledge of the program, right up to the result of the last test. Steps and test ideas are not specified in advance, and they may be recorded in great detail or not at all.

When a tester is working without a script, what can we expect to happen? If the test is memorized, or they have watched other testers perform the test, they may follow it as closely as they would if they had a recorded test script in front of them. If the test is not memorized, or has not been repeated so many times that it has become routine, we may see similar creative effects in testing as in improvised music.

In improvisational music, playing a euphonious note that fits with the ensemble and advances their discovery and engagement with the piece is important; in exploratory testing, performing some activity that fits with the project and advances discoveries and engagement with the product is important. In improvisational music, playing the right note is not so terribly important, but playing *a* right note is very important. If you wish to control the sound of the piece, emphasize scripting; if you wish to extend possible interpretations and knowledge, emphasize improvisation and exploration. In exploratory testing, our work is not as visible in the way music performance is, and we certainly can't hear what our tests are doing (unless we are testing music software.) Therefore, we have far fewer constraints when we improvise than our musical counterparts. We have less of a framework to work from, but more possibilities for discovery.

Automating tests is the strongest guarantee that they will be repeated exactly the same way, but like automating music, the lack of interpretation in execution can limit the results. A computer can only find the problems we predict and program it to find. Repeating scripted tests over and over can get boring, tedious, and may only feel like idea resolution, without the vital tension created by curiosity[Kohl2007]. At the other end of the spectrum, there is testing that is improvisational: exploratory testing. In the musical realm, electronic, or computer-assisted musical devices are fused with human efforts. This allows the musician to explore and create music that they would not be able to do completely on their own without the aid of tools. Similarly, in testing, we can use automation tools to help us work more creatively, and perform tasks that would be impossible without a machine helping us. [KohlM&M2007]

Skill and Skills Development

Both music and testing can be done easily by people without skill, but the perceived value of each is greatly enhanced by skill. Skill itself is enhanced by practice, the engagement of the performer, performance, knowledge of structures, and mnemonics that foster rapid learning.

In *This Is Your Brain on Music*, Daniel Levitin recounts considerable research into skills development in music. Levitin points out that skill and success in the music business are not strongly related; there are too many vagaries of timing, luck, and the whims of popular culture, and he acknowledges that expertise is a social judgment. For this reason, research involving assessments of musical skill has tended to focus more on technical achievement and innovation, and less on aesthetic appeal or popular acceptance.

Formal training (or its absence) is not necessarily associated with perception of musical skill. Many popular, skilled, and respected musicians, whether in popular music (Frank Sinatra, Louis Armstrong, John Coltrane, Eric Clapton, Stevie Wonder, Joni Mitchell, Irving Berlin), traditional music (Tommy Potts, Frankie Kennedy), or classical music (Gershwin, Mussorgsky and Beethoven) received little or no formal instruction.

Two key factors that do make a difference, according to Levitin, are emotional engagement and practice. The best students of music (and of other disciplines, according to related research are those that have practiced the most. Ten thousand hours of practice is required to develop world-class expertise[2]. Expertise in music, especially in music that is not heavily scripted, is associated with memory; and strength of a memory is related to the number of times that the original stimulus has been experienced.

The strength of the memory, and the associated development of expertise, is a function of emotional engagement—how much the user cares about the experience. To perform well, says Levitin, we have to pay attention and we have to care. More caring leads to more attention, and both caring and attention lead to neurological changes that mark experiences and memories as important.

Those who have not studied music may be surprised to find the great scientific, mathematical, philosophical and artistic energy that has been put into music over the years. In *Temperament: How Music Became a Battleground for the Great Minds of Western Civilization*, Stuart Isacoff mentions some of the people who were involved in solving problems in music: Pythagoras, Galileo, Kepler, Descartes, Newton, Huygens, da Vinci, Rousseau and others. [Isacoff2001] Music has a surprising depth into many areas of thought and study. Similarly, software testing is influenced by many disciplines, and has a surprising depth in many fields because of the vast number of technologies in use around the world. Music is not limited to learning the mechanics and rules to create and perform music, but is full of scientific, mathematical, social, political and artistic problems. Software testing is not limited to the execution of tests, and is also full of similar issues as music. We haven't learned enough about them all yet, and don't have the benefit of the many years and research that have been poured into music. We still have much to discover and learn about both.

---

[2] Levitin refers to Anders Ericsson, FSU.

Since the research that Levitin details on learning and music is consistent with learning in other disciplines, there are likely to be parallels that play out in testing. We propose the following hypotheses:

- We suggest that what we know about learning argues strongly for giving testers stimulating work that engages them, and argues against putting testers into situations where they simply repeat activities with which they are not engaged.
- Like music, developing testing skills requires development and practice. The software testing community could learn from musical counterparts as we develop exercises and practice software testing.
- Testing has very little of the aspects of physical performance, found in musical performance, that can obtain some benefit from rote repetition; there are few "muscle memory" skills in testing, but there are cognitive skills. Testing work that is boring or uncreative is less likely to be memorable, and thus less likely to lead to learning.
- Testing training that involves memorization of testing terms for the purpose of passing a certification test is unlikely to contribute much to the quality of software testing. In music, written theory exams don't begin until Grade 5; all testing and certification up to that point is based on performance. We hypothesize that the emphasis on technical terms found in current testing certification schemes adds little or nothing to the development of skill, just as the learning of musical terms contributes little to the quality of performance. As with musical performance, testing training that involves experiential learning, on-the-job training, coaching, and mentoring, will result in the development of skills.
- Schools of thought in testing ought to be encouraged, with more research into the differences and diversity of styles, genres and subgenres publicized for the learning profit of the software testing community as a whole. Avant garde, or cutting edge, experimental testing ideas and techniques should be encouraged, not written off. The resulting examples, cross-pollination and feedback loops would add more diversity to software testing.
- The manufacturing metaphor in software development is old, tired and often inappropriate. Even new variations like "Lean Manufacturing" do little to add to a software development field that is heavily based on design. Other creative, design-heavy fields should be explored, even artistic ones such as music.
-

Clearly more research is called for. The software testing discipline, like music, can be subtly complex and surprising. Merely taking an engineering or manufacturing view and trying to automate away the human labour-intensive side of music hasn't worked in music, and doesn't look like it's working in software testing either. There are more disciplines to learn from than engineering and manufacturing, and the musical field is full of ideas we can explore as we learn more about software testing.

Michael Bolton mb@michaelbolton.net
Jonathan Kohl: jonathan@kohl.ca

Notes

[BachRST] Bach, James, and Bolton, Michael, *Rapid Software Testing*. http://www.satisfice.com/rst.pdf

[Isacoff2001] Isacoff, Stuart. *Temperament: How Music Became a Battleground for the Great Minds of Western Civilization.* Random House, New York.

[Kohl2007] Kohl, Jonathan. 2007. *Exploratory Testing: Finding the Music of Software Investigation.* Self-published 2007 and Methods & Tools Magazine 2007 http://www.kohl.ca/articles/ExploratoryTesting_MusicofInvestigation.pdf http://www.methodsandtools.com/archive/archive.php?id=65

[KohlM&M2007] Kohl, Jonathan. 2007. *Man and Machine: Combining the Power of the Human Mind With Automation Tools.* Better Software Magazine http://www.kohl.ca/articles/ManandMachine_BetterSoftware_Dec2007.pdf

[Levitin2006] Levitin, Daniel J. 2006 *This Is Your Brain on Music: The Science of a Human Obsession.* Penguin Group, New York

[Pettichord2003] Pettichord, Bret.(2003-2007.) *Four Schools of Software Testing*. http://www.io.com/~wazmo/papers/four_schools.pdf