

T9: Text on Nine Keys

a Product Design History

Just as SMS was starting to explode in Europe, T9® Text Input made it easier for people to type on their mobile phone. Careful interaction design, repeated prototyping, and timely user research certainly helped make it an industry-leading product, but "putting the customer first" both ensured T9's commercial success and created some long-term challenges to overcome.

This article was written as a chapter for a book Scott Weiss proposed, *Case Studies in Mobile Design*, uniquely considering product design from the multiple perspectives of business, technology, and the user. The publisher never granted final approval for the book project, so the chapter is presented here in unabridged form.

Business Context

The primary focus for Tegic Communications, a small start-up frog in a large telecommunications industry pond, was to ensure success by meeting the needs of its primary customers, the handset manufacturers. At times this success required adapting Tegic's products to proprietary and industry standards even when the end user might have been better served (in the long term) by departing from those standards and establishing new ones.

Commercial Opportunity

The development of the Touch-Tone (DTMF) phone at Bell Labs in the '60s spurred a large body of work exploring how to use the 12-key keypad to combine telecommunications and computers in commercially beneficial ways. By the mid-'80s, additional research had investigated how to apply the by-then ubiquitous phone keypad technology for people with disabilities. The multiple letters and digit on each key needed to be disambiguated, either by the user while entering each character or by a company's server matching possible interpretations of the key sequence.

But the phone keypads were input only; any user feedback had to be by voice, synthesized by the computer at the other end of the line. A key element – immediate visual feedback to the phone user – was not available until mobile phones emerged with multiple-line text displays.¹

Around that same time, Short Message Service (SMS) was designed into Europe's new digital wireless infrastructure (GSM, for Global System for Mobile communications). As carriers deployed the GSM network, their per-minute voice pricing model gave subscribers an economic incentive to send a text message rather than make a phone call. By 1995, general use of SMS was starting to take off in Europe.

In the meanwhile, three engineers were each involved in developing assistive technologies for people with disabilities. Martin King developed a lightweight input device for an immobilized person; it mounted on the frame of eyeglasses and tracked the position of the wearer's eye. Dale Grover had developed the custom ASIC chips used in King's eye tracking controller. Cliff Kushler had worked for Prentke-Romich, the leading manufacturer of augmentative communication devices, for almost a decade and had worked with the other two men before.



They started investigating the most efficient way to do text input using only eight eye positions or joystick directions. When they reviewed the earlier Touch-Tone research, they recognized the opportunity for commercial applications and how success there could fund their disabilities work – and formed the company that became Tegic Communications.

Market Alignment

By the time Tegic completed its first commercial product, email had become the #1 use of the Internet. Tegic promoted T9 to handset manufacturers and wireless carriers as “email anywhere”, asserting that it made sending email while mobile as easy as receiving it. The campaign also highlighted market research showing that the mobile professionals who desired wireless email also represented the highest ARPU (average revenue per user).²

Based on their disabilities research, Tegic's founders initially advocated an optimized letter layout that minimized ambiguity while typing words. But (as I and others predicted) there was little interest among handset manufacturers for whom the Touch-Tone ABC layout³ was already becoming the de facto standard for SMS-capable phones. Eventually recognizing that their

primary customer was the handset manufacturer rather than the end user, Tegic principals shifted their strategy towards offering a solution compatible with existing mobile phones.

A big part of the product plan was ubiquity – making T9 available across most mobile phones, PDAs, pagers, and other small communications or computing devices. The product marketing team recognized that ubiquity was necessary to assure consumers that even the minimal time they spent learning this new technology would be worthwhile; effectively, “learn once, use everywhere”.

PDAs had better input and display capabilities than mobile phones and were seen as an extension to the PC, which was already common in US offices and homes. With open development platforms and no radio certification requirements, PDAs had much shorter development cycles than handsets and they created markets for third-party software. As a result, T9 had appeared on three different PDA platforms by the time it was available on the first widely-distributed handset.

This timing created problems for the ubiquity plan because touchscreen PDAs supported many other alternatives for text input. T9 significantly improved text entry on the phone’s fixed keypad; but on a PDA touchscreen the main competitive advantage of T9 was mirroring a T9 phone, and that phone was not yet available. In addition, prices of add-on products for the Palm were depressed (by all of the hobbyist shareware) below Tegic’s cost of retail distribution and customer support. Combined with other market shifts around 1999, T9 disappeared from new PDAs just as it began appearing on most new mobile phones.

Getting the Word Out

Because the direct purchasers of T9 were those responsible for handset design and feature requirements, rather than the end users, Tegic directed most of the awareness and marketing efforts towards the manufacturers and carriers and other influential people in the wireless industry. Third-party validation and awards⁴ helped “cross the chasm”⁵ with the large telecommunications companies.

As T9 began appearing on more handsets, the popular press and technology columnists also began reporting on the feature. The tireless PR staff attempted to keep up with the columnists to ensure that they were not misquoting Tegic literature or misunderstanding the technology⁶ which would incorrectly set expectations for new users. Magazines such as “What Mobile” soon included T9 in their profiles of new handsets and established it as a must-have feature.

At least one opportunity for raising consumer awareness was missed after Tegic was acquired by AOL Time Warner: Around the time of American Idol’s second season, before its popularity and ad rates skyrocketed, main Idol sponsor AT&T Wireless may have been amenable to a cooperative T9 education campaign. A simple 30-second demo by Ryan Seacrest showing how to text the word ‘VOTE’ could have done wonders for T9 use in the US.

Target Audience

Because it was cheaper in Europe to send a text message rather than make a phone call, its young adults became the key demographic for SMS, as they were the most price-sensitive and socially-oriented. In some countries, mobile messaging benefited teenagers otherwise constrained by social norms in public and at home.

Young mobile users tended to be technologically savvy and dexterous enough not to be put off by the demands of multi-tap typing. Their use of “txt slang”, which adapted chat room shorthand and radically abbreviated words to fit within 160-character message limits, reduced the pain of multi-tap by reducing the necessary number of characters. In turn, learning this lingo has reinforced the use of multi-tap.

Many young people therefore resisted using T9, either “out of habit” or “on principle”, even though their slang terms and abbreviations were easily added to T9’s dictionary. For example, quoting from assorted message boards:

“I was born a multitapper, and I’ll die a multitapper. I’ve tried switching to T9 and I’m never able to get the same speed I’ve developed just manually entering the text.”

“All right. I’ll give T9 another chance. It was the first option I looked up to turn off when it kept interjecting its suggestions. An invasive tool, for sure, but probably the deep psychological fear that a machine is going to start reading my mind. lol”

“I don’t see a reason NOT to use T9 if you can ADD words to its dictionary. I added EVERYTHING so all the curses and even an entire email address will be recognized.”

In contrast, research shows that the timing of multi-tap is problematic for older users.⁷ Late adopters of technology tend to be older as well. Since they are more likely to type full words rather than text slang, this audience seems well suited to T9 use. In addition, replying to email while mobile does not entirely excuse the sender from following email etiquette, including using correct grammar and properly accented words (another T9 benefit) instead of abbreviations and emoticons.

General literacy is another consideration, particularly in emerging markets, since key disambiguation systems that use dictionaries rely on written language standards and the spelling skills of users.

But to ensure commercial success, Tegic’s initial target audience had to be the decision makers within the manufacturer and carrier organizations. The goals of that audience included increasing SMS messages sent, subscriber satisfaction, and handset reliability, even while minimizing the overall device costs, including memory resources and engineering effort.

Prototypes

One of the first word disambiguation prototypes was a DOS program which responded to a data stream coming from Martin King's eye-tracking device.⁸ The block cursor on the screen moved among a 3 by 3 array of squares to indicate the eye direction of the user, and word matches appeared in a list.

I began working with the founders on the next prototype, which was a Windows simulation of a mobile email device. The simulation included up to 9 characters per key in an optimized letter layout, and allowed both ambiguous single-key and explicit two-key input. The keypad appeared a bit daunting even though the system promised excellent typing performance once learned.

We implemented the first mobile prototype on the Palm Pilot, after the PalmOS development kit was made available and it became a hit with hobbyist programmers. This prototype helped make things happen for the start-up company, for two reasons: First, the handheld form factor was closer to the experience of using a mobile phone, helping the manufacturers to understand the potential for this new text input method; second, by this point Tegic's executives understood that the optimized letter layout was not acceptable for a commercial product and we started using the standard Touch-Tone ABC layout.



But the prototype software, ported from the PC, was too slow for the Palm's 16 MHz processor and not robust enough to handle different people changing different parts of the program concurrently. So when a prospective customer requested an adapted Windows version for testing, I started to rewrite the internals too – both to ensure acceptable response times on mobile devices and in anticipation of a product release some day.



Then Texas Instruments licensed a commercial version for their new PDA, the Avigo 10, and the race to complete a release-worthy product was on. An old 8-bit processor, a proprietary OS, and a bug-ridden development kit gave us⁹ our first sense of what porting software to mobile devices was going to be like. And creating five European language versions from scratch in only eight months gave us our first sense of what we had to do to deliver for our customers.

Samsung signed up that year for the first mobile phone version of T9, in Korean and English. But for most phone manufacturers the PDA was still too different from their handsets to determine how T9 might work on a device with a few text display lines and even fewer function keys; and it would be many months before the first T9-enabled phone was released and could be used as a demo. So, for another year or so (until it became untenable), we created a custom prototype for each manufacturer – a fully operational Windows-based simulation that used a picture of one of their existing phones to show how T9 features could be mapped to its keypad and display.

Through this iterative prototyping process, we adapted our interaction design to the constraints of early SMS-capable handsets, and T9's features and appearance were greatly simplified as a result.

Some of the Windows-based prototypes also served other purposes: For Tegic's internal QA testing; for customer project teams to cross-check known T9 behavior against their handset integration work; and for marketing demos. The last group included the "Big Phone", an eight

foot tall simulated mobile phone for trade shows featuring a TV for the LCD display and hand-sized functional keys. Using it, we were able to demonstrate Alphabetic input and introduce our new Chinese input system.

Design Challenges and Compromises

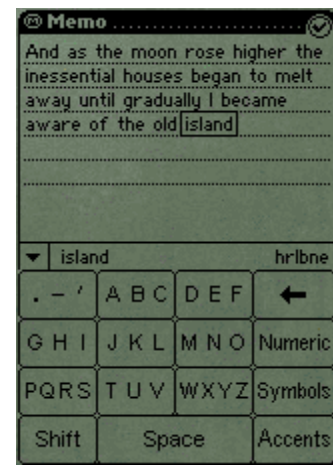
Technical Constraints

The display constraints and limited set of keys on the early mobile phones initiated a number of design compromises that proved difficult to reverse later.

For example, since its inception T9 was intended to support simultaneous interpretations of user input; that is, showing the numeric (e.g., phone number), two-key or multi-tap, partial word, and completed word interpretations of each key sequence. This made mode switching unnecessary or infrequent.

The PDA versions of T9 were able to seamlessly integrate exact spelling and ambiguous key interpretations, and the PDA displays were wide enough to show multiple word matches. But we decided to create secondary input modes to offer numbers, symbols, and other characters. This simplified the appearance of the main letter keys while still making those secondary characters easy to find and enter.

On early mobile phones with 12-character-wide displays, however, only a single word interpretation could be shown, inline at the text cursor and often without special highlighting. Users were required to cycle through three or more of the secondary input modes using a single mode key to obtain numbers and symbols and to spell new words.¹⁰ This became the lowest-common-denominator integration design for many generations of handsets.



Most mobile phones were also too constrained in storage and processing speed for the flexible software used for earlier prototypes and PDA versions, so Tegic's lead engineer, Ed Flinchem, directed the implementation of a T9 kernel appropriate to these limitations. The resultant code and compressed language dictionary required as little as 64K and was a key factor in the rapid adoption of T9 for most new mobile phones.

The repeatedly optimized dictionary format created some constraints on feature evolution, however, benefiting the manufacturer at some cost to the end user. For example, the compression technique produced some artifacts and prevented word completion on most words in the dictionary.¹¹

Discoverability and Training

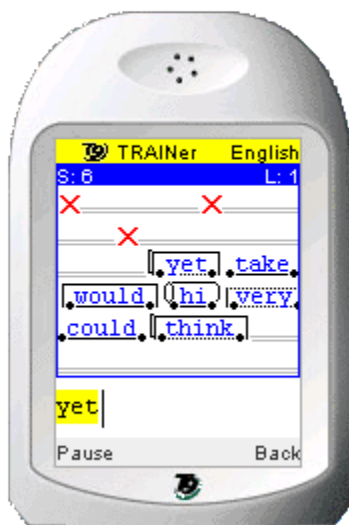
T9 has always faced a challenge in getting first-time users to the "Aha!" point where they understood and trusted its operation. We have observed that if another person showed them how T9 works, new users could pick up the basics in about 30 seconds. On the other hand, as seen in the earliest lab studies, unaided first-time users struggled for many minutes to understand what T9 was doing.¹² They tended to focus on fixing each letter instead of waiting until the complete word was entered.

It was apparent that users' prior experience with desktop keyboards and multi-tap input strongly influenced their expectations for how T9 should behave. For the first-time phone user, there was no analogous experience to draw upon when first attempting to type with ambiguous keys.

This was less of a problem for the PDA version of T9, where tapping accurately on each letter sufficed until the user realized that the system allowed less precision. The only unexpected behavior on the PDA version was that the system might accept the most likely word instead of the explicitly spelled word.

Ultimately, success depended on the user being able to learn or discover enough about disambiguation to use the T9 system. But most users never opened the manual¹³, and in most cases the device itself was not able to provide much educational information. Even in the late '90s, only PDAs like the Philips Nino provided sufficient program storage and an interface rich enough to support an on-board tutorial. Explaining how T9 worked was not a user interface element we required our mobile phone customers to implement, especially given the difficulty of doing so on a three-line text display and our reluctance to interrupt the user's input task with a quick how-to session.

Initially handset manufacturers understood that T9 was different enough from multi-tap to cause confusion and were willing to include in the retail box the "T9 Quick Tips" single-page flyer we designed.¹⁴ They were most effective in printed form and when referenced as the user was first discovering T9. But the wireless carriers often included another half-dozen flyers of their own in each box, blunting the impact of the T9 flyer.



Palm offered the "Giraffe" game with their PDAs which caused people to learn Graffiti unistroke handwriting in the process of playing the game. Encouraged by that example, we designed and eventually developed a Java applet called "T9 TRAINER" to introduce the basic concepts and features and to provide some practice time in a fun setting.¹⁵

The technical and marketing communications teams quickly faced a challenge in explaining how to use T9 on each phone, given the mushrooming number of models and the differences between implementations. Relying on generic instructions meant that users couldn't resolve problems with T9 on their particular phone model.¹⁶ Some of the manufacturers and carriers augmented the user manuals with FAQ lists on their web sites, but the instructions have often been generic there as well.¹⁷

Industry marketing also influenced user expectations, which in turn affected discoverability. For example, after T9 became a differential advantage for mobile phones a leading manufacturer started calling the feature "predictive text" generically. We were rather concerned by this because the system only disambiguated complete keystroke sequences and did not predict the completion of words.¹⁸ But, lacking a more marketable term than "disambiguation", Tegic's product management eventually accepted this de facto term describing the market niche.¹⁹

Our concerns were later realized, after PDAs and other mobile devices added word completion to their input methods and some misstatements by tech columnists and handset reviewers reinforced the expectation that T9 would predict the word you were starting to type, e.g.:

"Tegic's T9 software, which reduces the number of keystrokes by offering choices to complete a word as the first few letters are entered."

Guidelines vs. Requirements

It is ironic that, though T9 is a user interface feature, we really never had control over its user interface in any handset implementation. Recognizing that physical and software constraints would be an ongoing problem requiring intelligent tradeoffs, I compiled a list of user interface strategies to address new device and display capabilities, and we referred to them when manufacturers asked about novel situations. The head of our growing design and usability team, Christina James, developed a presentation illustrating the key user interface elements necessary for a successful T9 implementation, which became part of the training session for new customer engineering teams.

But the number of handset models quickly multiplied. Tegic's design team no longer worked with a manufacturer's engineering team after the first training session, and the engineers went on to implement subsequent models. Though the lead T9 product manager, Lisa Nathan, created a user interface element checklist to ensure that key elements were addressed satisfactorily, not even the checklist could be applied when Tegic wasn't informed of a new handset before release. As a result, manufacturers occasionally produced a mobile phone with a T9 implementation that was inferior to those on their earlier handsets.²⁰

Though the issue was debated from time to time, the product team held to the decision that only the contractually required elements would be enforced and that no handsets would be singled out as having better T9 implementations than others. Whereas a few software platform vendors had created some ill will in the mobile device industry by attempting to control much of the user experience, Tegic took the laissez-faire (or "friendly partner") approach – recognizing that, once it was integrated, intelligent text input became a less important part of increasingly complicated and rapidly evolving digital devices. The manufacturers were allowed to integrate T9 in any way that best served the needs of their device's overall user experience.²¹

The downside of this flexibility, and subsequent success and ubiquity, is that there was little standardization between devices. Most manufacturers strived for UI consistency within their handset series, and the user benefit of a T9 standard across competitors' devices did not supersede it.²² For example, the Next Word function was necessary to select alternate words; it appeared on at least four different keys across devices and some manufacturers refused to ever label it despite our requirements.

In some cases, consistency even within devices was problematic. For example, the Create SMS Message applet usually had a richer text entry interface than the phonebook fields. And the microbrowser often established entirely different behaviors from the rest of the handset – forcing the mode toggle to be on the left softkey when it was on the right softkey everywhere else, or using long-press on each key for site bookmarks when that action inserted a digit in the messaging application.

As T9 was the first commercial disambiguation system, it was introduced on handsets that already had standardized on multi-tap. It was unrealistic to expect the manufacturers or their users to replace multi-tap immediately, and in fact we initially recommended that T9 should not be the default input mode. Users found each new generation of digital phones complicated and frustrating enough without forcing them to understand T9 while merely entering a name into the phonebook. We wanted them to try T9 when they were in a "discovery" mindset and amenable to learning something new.

But, again, this early accommodation remained a default implementation long after the concern passed. A few handset models did not allow T9 mode to be used at all while entering phonebook data or Instant Messaging buddy names, creating puzzling inconsistencies for the user.

Even the T9 program icon became entangled by handset capabilities and manufacturer preferences. Intending it to become the ubiquitous mobile ingredient brand equivalent to “Intel inside” and a quickly-recognized symbol of full-word entry, Tegic required the T9 icon to appear whenever an entry field was in T9 mode. But Nokia insisted on their “fast writing” indicator for consistency with their own UI conventions, and Openwave’s early WAP browsers implemented a default input mode label of “WORD”. In addition, many handset status displays overloaded the multi-tap mode indicator “ABC” with shift state, e.g. “Abc” for single-caps. Since the letter and digit in “T9” couldn’t readily show shift states, the handset designers would request an alternative mode indicator.²³ Given all this, Tegic’s product marketing team found it difficult to establish a consistent cue for T9 Text Input, eventually relenting and allowing either “T9” or “WORD” to indicate the entry mode.

User Research

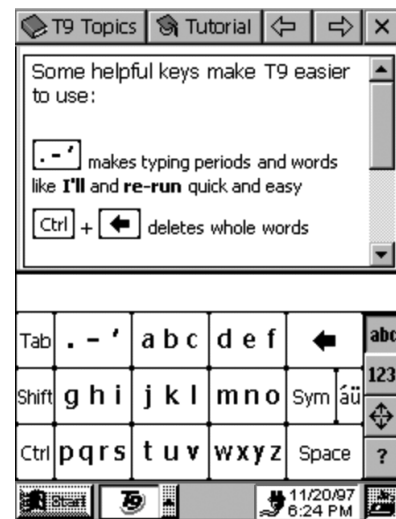
In the beginning, research efforts were focused on addressing the communication needs of the physically disabled. After reviewing the earlier Touch-Tone system work and other research into disambiguation and word prediction, Tegic’s founders experimented with different letter combinations, visual presentations, and other optimizations to reduce cognitive load and increase keystroke efficiency. In 1995, they presented the results of their work to the disabilities community in the form of a prototype system named JustType®. The founders soon realized that there were commercial applications for the same technology, particularly on one-handed devices like phones and pagers; in some ways, the size constraints imposed by mobile devices cause “disabilities” in all people.

During Tegic’s start-up phase, as with most small software companies, survival took priority over usability. But we looked for user feedback whenever and however we could.

Philips Electronics wanted to differentiate their new PDA, the Nino, from its Windows CE competitors and chose to bundle a version of T9. I took the opportunity to check the usability of some of the user interface elements and of its on-board T9 tutorial.

Lacking a formal lab, and recognizing the inherent contextual aspects of mobile device usage, I improvised by bringing the user tests to where the users were. In the first case, visitors to an annual University of Washington computer fair were invited to stop for 15 minutes or so and “try out a new text input method on a new PDA.” A year later, for the color display version, students at Seattle Pacific University signed up for half-hour slots as they walked between classes in the Science and Engineering building.²⁴

We also took advantage of other organizations’ usability labs. A visiting professor took the lead for our first formal study hosted at the UW’s new LUTE Lab, which examined the problems experienced by the first-time T9 user. Though ultimately successful, the study did reveal issues with outsourcing the testing of a mobile device product with subtle features, trade-offs such as the time it took to educate the test administrator and results analyst, and challenges such as synchronizing prototype completion with lab availability.





Since there wasn't a lot of literature on the subject of mobile device user testing, we had to experiment before formalizing our usability program. For example, phone simulations running on a PC had to be modified so that they were effective when operated with an overlay touchscreen. The modifications included larger buttons, audio or haptic feedback, and button activation based on original touch location rather than release location. We also found later that using a car dashboard mount to lock the mobile device in place was useful for videorecording the keypad and display over a study participant's shoulder.

Once T9 became available on the majority of new phones, we were under some obligation to carefully evaluate new features and user interface variations to ensure that the product was becoming more effective for the user and not merely laden with additional features.²⁵ Most new features and UI changes were subject to expert review and formal usability testing. For instance, a number of usability studies helped refine the behavior of T9's unique "Smart Punctuation" feature (on the 1 key) to make it as easy as possible to enter URLs and the like. Similarly, Tegic's phrase-based Chinese text input system was guided by iterative design and testing.

There were a number of improvements that never made it to commercial release because the user benefits were unclear or insufficient given the added interface complexity or code required. For example, we discovered through testing that certain visual presentations deterred novice users from discovering T9's full-word disambiguation model, as they quickly fell into a pattern of selecting the desired letter after each keypress as an analogue to multi-tap. And our studies made it clear that we would not be able to devise a simple "automatic help" mechanism (like the oft-scorned Office Assistant "Clippy") that could correctly infer, without knowing the word being entered, when a novice user was having trouble and what kind of help to offer.

Languages

Much of the initial feedback came during language testing. We sought out native speakers in each language to review the system during development to confirm that the process of text entry felt "natural" to them and to ensure that the vocabulary covered the common words and grammatical forms.

Each language, and society that speaks it, carries implicit expectations for the behavior of text entry. Often we had to uncover what those expectations were, as they could be based on secondary school training, desktop PC conventions, or experience with rudimentary multi-tap systems previously offered by phone manufacturers.

From that emerged an issue of user representation: To validate the new language input system, study participants had to be fluent in the target language, of course, but given limited testing resources was it better to recruit typical phone users or skilled linguists? Many of the questions we needed to answer were regarding characteristics of the language – e.g., did the keypad layout make intuitive sense given the language's writing system; were all of the grammatical tenses fairly represented in the dictionary; did entry of words or phrases divide up logically? But *typical* phone users remembered very little from grammar class and could not help with these kinds of questions. They typed using a very small subset of the language, often augmented with texting abbreviations and slang that didn't follow the rules of their native language. On the other hand, we were designing the interface for the casual user and wanted to test it with people of all language skill levels, not just PhDs.

Performance

Ultimately, the point of a mobile text input system is to make it possible to type as fast as one can think, or failing that, at least getting as close to desktop touch-typing speed as possible. The early 10-key optimized layout T9 prototype suggested 55 words per minute (WPM) was possible,²⁶ which is within the range of a professional typist. For comparison, another study²⁷ showed that desktop computer users only average between 19 and 40 WPM.

Silfverberg et al published a paper²⁸ that sought to define a theoretical model for performance based on Fitts' Law. In it, they compared multi-tap and two-key input methods against T9 single-key input and used experimental data to determine the formula coefficients. The model predicted a (one-thumb) "expert" entry rate of 40.6 WPM for T9, but the authors also suggested a downward adjustment in performance based on the projected need for visual verification at the end of each word.

Concurrently, Dunlop and Crossan²⁹ proposed a keystroke-level model to estimate the performance of multi-tap and single-key input methods, resulting in 14.9 and 17.6 WPM respectively, again assuming an "expert" user.

In 2001, Tegic's Christina James and Kelly Reischel published the results of a follow-up study.³⁰ They measured the actual performance of experienced and novice users on both multi-tap and T9. Interesting results included the fact that, while novice users typed at about the same speed using either input method (around 8 WPM on average), experienced multi-tap users were not significantly faster than that—whereas experienced T9 users more than doubled the average rate (20.4 WPM overall). The authors proposed a number of considerations that could account for the differences between the models and the observed performance.³¹

Since then, additional university student research papers and competitive studies have appeared. MacKenzie and others developed metrics such as keystrokes per character (KSPC) to better compare input methods. Unfortunately, many of the third-party and university studies put great effort into equipment setup, participant preparation and training, etc., only to time the entry of 2-3 short phrases and then base performance estimates on that. Unfamiliarity with the keypad layout, device, and input method may unfairly lower its predicted performance if a novice is never allowed the chance to get comfortable with it.

Audience Acceptance

As might be expected from the challenges noted above, the acceptance of T9 often depended on how each user discovered or was taught its operation and use. T9 is not "intuitive" in the common use of the term,³² due to the fix-each-letter presumptions many people arrived with. New users needed to be helped towards a different understanding; if that didn't happen quickly, they were just as likely to discover how to turn it off. Paradoxically, the ease with which people accepted the new understanding once they were *shown* how T9 works indicates that it was not a particularly complex concept.

Other expectations may have come into play. Anecdotes from some first-time users indicated that, once they got past the idea of entering the entire word, they were then stumped because "it doesn't always give me the word that I want." We tried to enforce use of a standard icon or the word "Next" for the Next key label, but in user tests we observed a conceptual barrier for some people such that they didn't think of looking for a key or function that might give them another word choice.

Larger handset displays with word lists to scroll through have since helped eliminate that hurdle. But that earlier discoverability problem raises a few general questions about mobile devices and how their applications are designed: Are they so complex that some people have no hope of figuring them out and give up at the first sign of trouble? Or are expectations of quality so low that some people just assume that there's a bug and never return to the feature?

Discovery issues aside, a number of surveys and diary studies in various parts of the world indicated that people who used T9 sent more text messages than those who didn't. This kind of bottom-line motivation helped carriers ensure that a product like T9 was included in virtually every new mobile phone.

According to another study, once users "got" T9 they didn't want to go back to multi-tap. Other anecdotal evidence confirms this, including these postings on various message boards:

"I used to multitap (the "regular" way), but I switched to T9 and don't think I can go back."

"I would have to concur ... that T9 is much faster... It took me a couple days to get used to it, after using multitap for years. Allows you to hit the button that has the letter, once, and it will make the word for you; for some words you may have to hit Next to find the word that is the same buttons pushed, but it works like a charm."

"T9 is good and faster – once I learnt how to use it, I stuck to it."

"I think texters often ignore T9 until someone shows them how it works. It's not something that you try to learn by yourself coz it looks confusing at the beginning."

"I love the Nokia implementation of T9... I am probably at about 30 words a minute with T9... whereas regular [multi-tap] is slower than hell."

The other audience – the handset manufacturers – demonstrated great acceptance of T9. This was due to a number of factors in addition to its feature design, such as the efficient size and engineering quality of the software, reliable technical support, broad language support, and Tegic's patent portfolio.

Epilogue

Nuance Communications (which acquired Tegic from AOL in 2007) has estimated that over 3.5 billion devices have shipped with T9, representing more than two-thirds of mobile phones worldwide and enabling mobile input of over 70 languages in 15 scripts.

Subsequent versions of T9 added word and phrase completion, dual-language support, and other useful features. Our "SloppyType" invention,³³ conceived for touchscreen QWERTY keyboards, became the underlying technology for many mobile text input variations and now supports spelling correction for advanced T9 implementations.³⁴ The new T9 Nav® search and discovery tool harkens back to the early ability of T9 to offer simultaneous numeric, partial word, and completed word interpretations of a key sequence.

Ironically, though many feature phones now have display and storage capabilities better than those of the early PDAs and their keypads have acquired dedicated function keys, handset project teams still do not take advantage of all of T9's capabilities when implementing the user interface.

What will the future bring for T9? Most new mobile phones continue to be built with 12-key keypads,³⁵ though the iPhone has finally popularized the idea of a touchscreen-based

“convergence device”. T9's word-based disambiguation and correction can improve speed or accuracy for input devices besides phone keypads, a subject I've explored in another article, [“Nothing New Under the Thumb”](#).

Nuance recently announced a third-party developer program to allow greater access to the T9 technology suite. It seems that Tegic's early marketing phrase calling T9 the “Global Standard for Mobile Text Input” will be true for years to come.

ENDNOTES

-
- ¹ Around the early '90s, e.g., Nokia 101 (1992, pre-GSM), though pagers and “digital assistants” with multi-line displays preceded them.
- ² Eventually, mindshare of that high-value target demographic was captured by RIM’s Blackberry.
- ³ Updated to include Q and Z in their alphabetic positions on the 7 and 9 keys, respectively.
- ⁴ T9 was named Washington Software Association’s “Best Consumer Product of the Year” in 1999 and Mobile Insights’ “Best Software Application of the Year” in 2000.
- ⁵ Moore’s *Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers*, 1999.
- ⁶ Typically, “it predicts the word for you after just a few keypresses.”
- ⁷ Jastrzembki, “The Model Human Processor and the Older Adult: Validation and Error Extension to GOMS in a Mobile Phone Task”, 2006; available [here](#).
- ⁸ Another early input device prototype was “TippyType”, an 8-way joystub placed on the end of a handheld stick, similar to the “hat” switch on a game controller joystick. Since the game console market was much smaller than the mobile phone market and had a complicated value chain, Tegic never developed this idea further.
- ⁹ A team comprised of the founders and a rotating set of contract developers and linguists, making most of the “Classic Mistakes” listed in McConnell’s *Rapid Development* along the way.
- ¹⁰ Competitive handset studies and anecdotal evidence have suggested that the difficulty of mode switching on some phones can become a real annoyance for T9 users.
- ¹¹ Even though research showed that word completion did not really increase ambiguous input performance, users perceived it as beneficial and customers often asked for it.
- ¹² Far longer than the typical user is willing to try on his or her own before turning it off entirely.
- ¹³ Unless they got stuck; certainly not to casually discover the 1-3 pages about text input among the 100 pages of features and safety warnings!
- ¹⁴ Refined through user testing in a number of languages.
- ¹⁵ Manufacturers and carriers can offer the game as a download from their web sites, but most have not chosen to pre-install it on their handsets.
- ¹⁶ e.g., identifying the unlabeled Next key, or confirming that (due to a bug perhaps) adding a word to the T9 dictionary worked from the SMS applet but didn’t work from the Instant Messaging applet.
- ¹⁷ The tutorials and instructions at the T9.com site have since been improved to address some of these challenges.
- ¹⁸ It certainly didn’t help our efforts to establish the “T9” brand either.
- ¹⁹ I have since wondered whether it would have been more effective to use a phrase like “dial-a-word” to introduce T9 to first-time users, especially for the U.S. audience with whom the concept would be familiar, e.g., 1-800-FLOWERS.
- ²⁰ To the detriment of many, the poor implementation was also (as often as not) on a mass-market “free” phone that became the most widely distributed!
- ²¹ Or that matched the skills and attention of its engineers.
- ²² When ETSI began specifying standard key layouts for multi-tap and predictive text input, it was unclear which of the major proprietary layouts would win out, or if all of them would be accommodated.
- ²³ We advocated a separate up-arrow icon to indicate unshifted, single-caps, and all-caps states; but some handsets had no space for an extra icon on the status line.
- ²⁴ Though one could argue that the surrounding chatter of people and poor overhead lighting generated its own usability problems, and non-reproducible study results, the resultant issues (like screen reflections and poor LCD contrast) did represent likely problems during actual use that should be designed-around if possible (such as not depending on subtle colors and font attributes for meaning).
- ²⁵ The lack of extra function keys or cascading menus certainly discouraged *secondary* features.
- ²⁶ Using a standard 10-key keypad, a professional typist reached an estimated 55 WPM in a repeated-entry test. But formal studies with multiple participants were never performed on the optimized keypad layout.
- ²⁷ Karat et al, “Patterns of entry and correction in large vocabulary continuous speech recognition systems”, 1999.

²⁸ Silfverberg et al, “Predicting Text Entry Speed on Mobile Phones”, 2000.

²⁹ Dunlop and Crossan. “Predictive Text Entry Methods for Mobile Phones”, 2000.

³⁰ James and Reischel, “Text input for mobile devices: Comparing model prediction to actual performance”, 2001.

³¹ The consideration that caused the most animated debate, however, was the difference between an “expert” user and an “experienced” user. (Moral: Choose your words carefully when testing someone else’s theoretical model.)

³² For a better understanding of the term, see Jared Spool’s [blog posting](#) on the subject.

³³ Robinson and Longe, “Keyboard system with automatic correction”, US Patent 6,801,190 and others.

³⁴ Now marketed as XT9 Smart Input.

³⁵ Particularly mass-market phones for pay-as-you-go and emerging markets.