

2. Select team members based on what is good for the firm and not out of duty towards them – because, for example, they have been overlooked for another job or have been with the firm the longest.

3. Make sure you know your staff well enough to be able to determine how they'll work within a group. Get feedback about their competencies and team-working skills from other staff, managers and, if possible, through regular discussions with the persons concerned.

4. Not every talented person is a team player. Good team players have good listening skills, are cooperative, can represent their own area of expertise, see the project's success as a group effort and are flexible and secure. They also have good meeting skills, are able to identify and solve problems, are good communications and are able to give and receive feedback.

5. Choose the team leader carefully. Technical competency isn't enough to create a leader. A leader must have good team-motivation skills and be able to produce results.

Many other factor, apart from specialist skills team-working abilities, go into choosing the right team. For example, a project may need staff who are skilled at dealing with a difficult client. Or, if there is a tight deadline, those who work well under pressure will be chosen. Inept teams, especially at senior levels, can do irreparable damage to a company's brand, product line, customer relationships and share value.

*Scientific supervisor: Rozum M.I.,
Senior Lecturer*

UDC 311.21:004 (043.2)

Suprunenko V.R.

National Aviation University, Kyiv

HOT PROGRAMMING TRENDS

Preprocessors over Full Language Stacks. Some time ago people created new programming languages and along with it they had to build tools for turning code to bits flew through silicon wires. Later someone guessed they could reach result on the work that came before. Today people who have clever ideas simply write a preprocessor that converts code of new standards into something old fashioned code with a rich set of libs and APIs.

JavaScript MV* Frameworks over JavaScript Files. It wasn't long time ago when coders learned to write JavaScript to show an alert box or to validate email input field for presence of "@" sign. Today HTML/Ajax applications are so complicated in their structure that few people start from the very beginning. It's much simpler to adopt developed framework and write some of middleware code

to provide your business logic. Those are merely the Web apps. There are also a number of frameworks set to offer development for different platforms from the smartphone/tablet world.

CSS Frameworks over Generic CSS. Nowadays Web pages are built with so high complexity today that it became impossible to create a file using just simple commands. Here is the entry point for CSS frameworks like SASS have found sturdy base. They stimulate laconic and stable coding by offering real variables, nested blocks, and mixings as programming constructs.

SVG and JS on Canvas over Flash. Flash technology has been unbelievably popular for years, but everyone loved only the results. Nowadays the JavaScript layer knows how to do much of the same, browser producers and web developers are hope to see the end of Flash. Better integration with the Document Object Model born from new tools like SVG (Scalable Vector Graphics) is obvious for them. Markup language with scalable graphics introduce one big heap of tags (often they are easier for developers in usage),

Single-page Web Apps over Websites. New Web apps are front ends to huge databases charged with content. When Web application want to receive some new information, it gets it from the database and fill it into the current template. There is no need to mark up the data with all the Web extras needed to build a Web page. The data layer is absolutely separated from the presenting and formatting layers. Here, the growth of computing is another reason: responsive-designed single Web page which works just like an application.

Mobile Web Apps over Native Mobile Apps. Having great idea for mobile app stores you could rush off and created splatted versions for Android, Windows 8, iOS, BlackBerry OS, etc. Each of these platforms needs a separate team using different programming languages. Another way is to build one app using HTML/CSS and to run it through all the platforms. Now that the HTML layer is getting much faster, this approach can lead to native apps being better and more complicated apps.

Web Interfaces over IDEs. It is time for the IDE to be substituted by the tools based on the browser that let you edit the code, usually of a working system. It allows coding and editing on fly. These systems do not suggest cool debugging, and we may consider it as something dangerous to rewrite the code from production, but this idea has legs.

NodeJS over JavaEE and Ruby on Rails. The server world has always flourished on the model of threads that let the operating system pander any wildered or inefficient behavior by programmers. Then along with JavaScript callback model of programming came Node.js, and the speed of code execution really grew up. Suddenly the over cost of creating many threads was revealed and Node.js took off. The Node.js world offers stability between server side and browser. The same code may run on both sides easing developers work by duplicating functionality.

From the information considered it is easy to make a conclusion about moving vector of the whole IT world. We are migrating to the net either users or developers. It is a good tendency while developing in WEB and for WEB is unbelievably comfortable, easy and fast to deliver useful products to the users throughout the world. Although, people are still speeding up world of software. Education becomes shorter till developers learn new technologies and instruments on-fly.

*Scientific supervisor: Babiy H.V.,
Senior Lecturer*

UDC 656.7.072 (043.2)

Suvorova K.O.

National Aviation University, Kyiv

DEVELOPMENT OF LOW-BUDGET AIRLINES IN UKRAINE

Air transportation is the most important and integral part of the economy of the developed countries. There is a whole range of systemic problems despite the positive dynamics of growth in the domestic air transportation in Ukraine today. The model, used by the national airlines today, is not suitable for both internal and external airline markets.

The end consumer can't afford paying a lot of money for tickets, offered by the airline. This is a well-known fact, that in the period of economic crisis, inflation affects the income of the consumers significantly. Maximizing the profit of airlines should be achieved at the expense of affordable prices on airline tickets that will increase passenger traffic.

Low-cost airline has built a completely new business model, based on high productivity and low costs.

The significant factor in the low-cost airline is the efficiency of business, which is higher than traditional carriers. They offer tickets at low prices. That is why the passenger traffic increases. This allows airlines to get more aircraft hours, more profit with the less number of employees and less spending on each passenger seat. The main factors to achieve this effect are: using the modern and young fleet (new aircraft consume less fuel and less likely to break); excluding the cost of optional elements such as flight catering, less staff; sales of airline tickets over the Internet.

Another feature of the air transportation market of Ukraine is that, the model of air transportation is not always effective on domestic routes. The airline's performance depends on the end-users, i.e. passengers, because they define how and where to fly. Modern passengers in a much greater degree are differentiated according to their needs, compared to passengers of the last century.